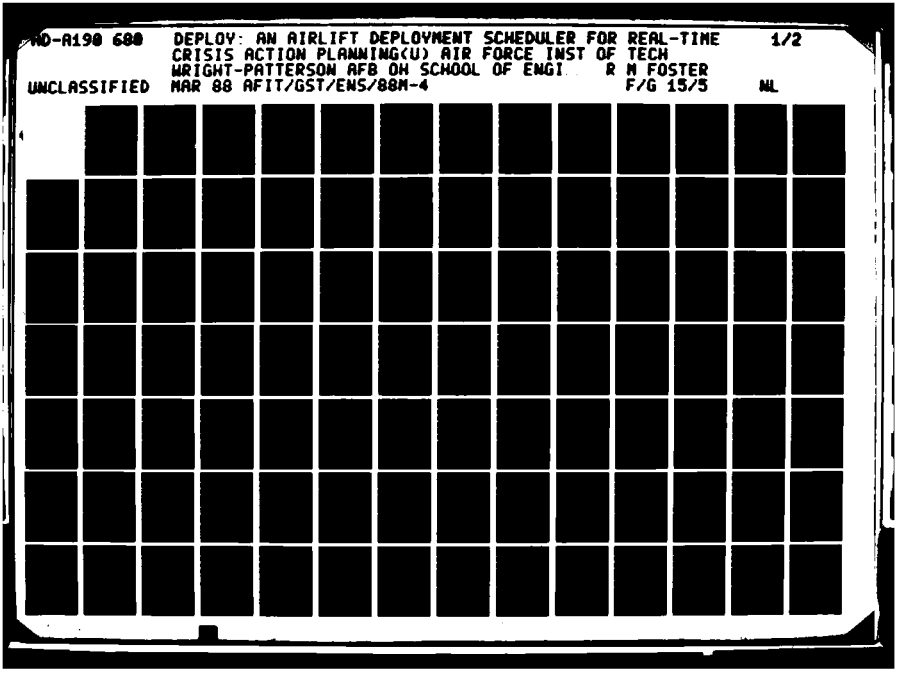
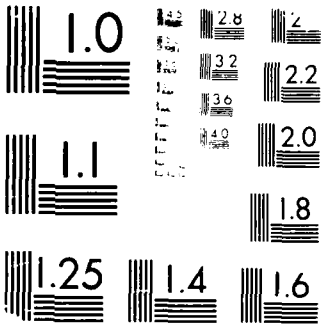
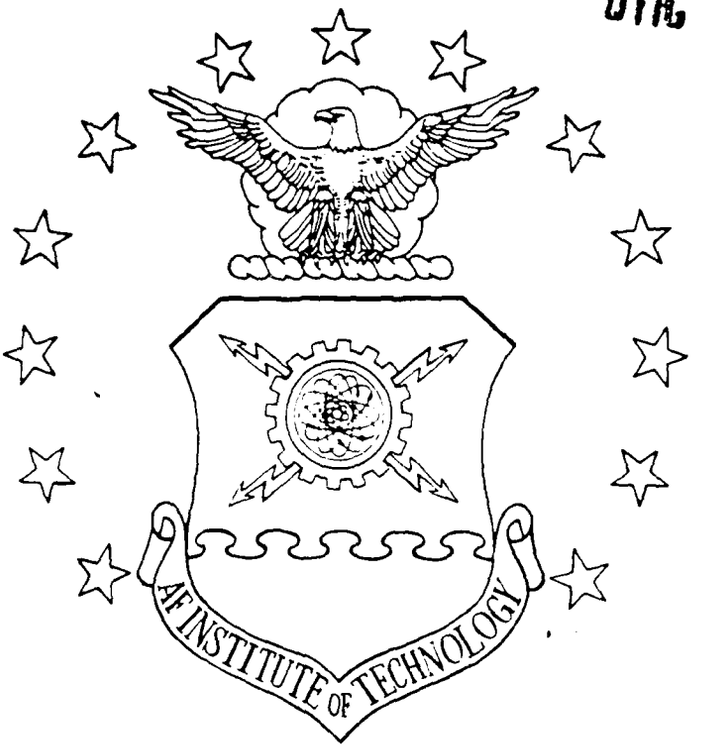

AD-A190 680
DEPLOY: AN AIRLIFT DEPLOYMENT SCHEDULER FOR REAL-TIME CRISIS ACTION PLANNING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI R M FOSTER
1/2
UNCLASSIFIED
MAR 88 AFIT/GST/ENS/88M-4
F/G 15/5
NL



MICROCOPY RESOLUTION TEST CHART



DEPLOY

AN AIRLIFT DEPLOYMENT SCHEDULER FOR REAL-TIME CRISIS ACTION PLANNING

THESIS

R. Mike Foster
Major, USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GST/ENS/88M-4

DEPLOY

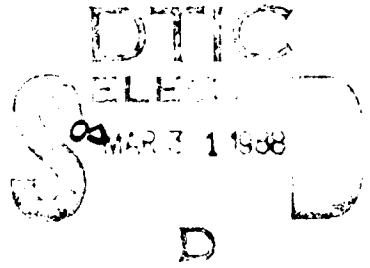
AN AIRLIFT DEPLOYMENT SCHEDULER

FOR REAL-TIME CRISIS ACTION PLANNING

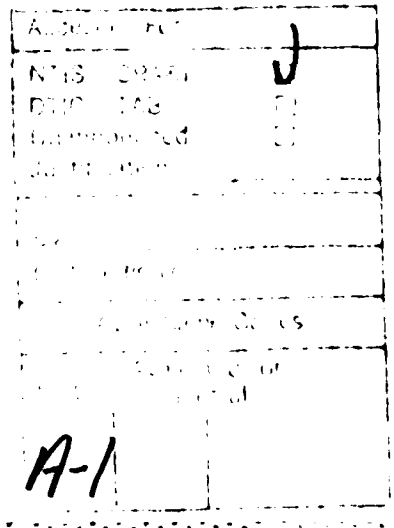
THESIS

R. Mike Foster
Major, USAF

AFIT/GST/ENS/88M-4



Approved for public release; distribution unlimited



AFIT/GST/ENS/88M-4

DEPLOY

AN AIRLIFT DEPLOYMENT SCHEDULER FOR REAL-TIME CRISIS ACTION PLANNING

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

R. Mike Foster, B.A., B.S.

Major, USAF

March 1988

Preface

The purpose of this research was to develop a methodology for automating the planning of aircraft deployments from the United States to Europe. The immediate need is the deployment planning of C-130s, but the methodology, with modification, can be applied to other deploying aircraft such as fighters and tankers.

The DEPLOY program reduces deployment planning time and provides a means to store plans for future use or modification.

I would like to thank my faculty advisor, Lt Col Thomas Schuppe, for his guidance and patience. I would also like to thank my reader, Lt Col John Valusek, and my friend, Capt Joe Miller, for their suggestions and criticisms.

I would like to express my deepest appreciation for the patience and understanding of my wife, Mary. Finally, I would like to say a special thank you to my daughter, Julie, for letting me use 'our compooter' for this thesis.

Table of Contents

List of Figures

List of Tables

Abstract

Preparation for a war in Europe would include the deployment of C-130 tactical airlift aircraft from their home bases in the United States to beddown locations in Europe. Methods currently used to plan such deployments would require as long as two days to provide a complete deconflicted deployment schedule.

The purpose of this study was to automate the deployment planning process. A review of literature concentrated on recent theses which studied airlift problems and existing deployment planning software.

An interactive program was written, based on the concept of using departure time from refueling choke points as the mechanism for regulating aircraft flow. A deployment flow plan for all active duty C-130s stationed in the U.S. can be completed, using this program, in approximately one hour.

DEPLOY

A Microcomputer-based Airlift Deployment Scheduler

For Real Time Crisis Action Planning

I. INTRODUCTION

Military Airlift Command (MAC) is tasked with the worldwide airlift of United States military forces in support of national security objectives as defined by the National Command Authority. This tasking includes both strategic (between theaters) and tactical (within theater) airlift of forces. The strategic airlifters include the C-5 Galaxy, the C-141 Starlifter and the KC-10 Extender. These aircraft are assigned to active duty, Air Force Reserve and Air National Guard units located throughout the continental United States (CONUS). The Civil Reserve Air Fleet would also be made available to MAC for the airlift of troops and bulk cargo. The aircraft currently used for tactical airlift is the Lockheed C-130 Hercules. Active duty C-130s are permanently stationed in Europe, Japan, the Phillipines, Alaska and the CONUS. Two-thirds of MAC's C-130s are assigned to Air Force Reserve and Air National Guard units stationed in the CONUS.

In a European contingency, strategic airlifters will shuttle between Aerial Ports of Embarkation (APOEs) in the

CONUS and Aerial Ports of Debarkation (APODs) throughout Europe carrying troops, ammunition, trucks, tanks, food and other expendable supplies. C-130s will deploy from their home bases to theater beddown locations from which they will conduct theater airlift operations. They will deploy with maintenance personnel and War Reserve Spares Kits, thus having a limited degree of self sufficiency. Operating out of the beddown locations, they will haul material and personnel from the APODs and seaports located well behind the lines to airstrips and drop zones near the battle.

Deployments of MAC aircraft are planned by Hq MAC/XOSP located at Scott AFB, Illinois. The strategic airlift deployments, or flow plans, are planned with the aid of MAC's Flow Generator program, FLOGEN. FLOGEN is used during deliberate planning and contingencies to schedule strategic airlift flows to meet cargo and passenger requirements. FLOGEN runs on a Honeywell 6000 computer and is capable of processing up to 12,000 missions (10). FLOGEN uses a cargo requirement-driven algorithm, which makes it unsuitable for the deployment of a C-130 unit.

PROBLEM DEFINITION

C-130 deployments are currently planned by hand. It has been estimated that a full scale deployment of active

duty, Reserve and National Guard C-130s to beddown locations in Europe would take MAC/XOSP two days to schedule (12).

RESEARCH OBJECTIVE

The objective of this research is to develop a methodology for automating the deployment planning of C-130 aircraft from CONUS bases to beddown locations. The methodology should be useable for deliberate planning, real time crisis action planning and validating the feasibility of existing plans.

SCOPE

To meet the requirements of Hq MAC/XOSP, the methodology must be:

1) Responsive in real time, during the Course of Action Development phase of the crisis action system;

2) Useable on microcomputers they now possess (Z-150 or Z-248); and

3) Modifiable for use in other areas.

Input parameters would include unit and home station, the number of aircraft assigned to the unit, beddown location, routing, choke point(s), arrival/departure interval at choke point, movement priority by unit, and maximum number of aircraft allowable on the ground at enroute bases.

Output should include a printed schedule of aircraft movement, fuel requirements at all airfields involved, and the date and time each unit completes its deployment (unit closure)(8).

OVERVIEW of THESIS

Chapter II is a review of literature and software which address areas closely related to the research objective. Chapter III details the methodology used and the resulting program, DEPLOY. Chapter IV gives a demonstration of DEPLOY in an experiment to compare closure times of various deployment options. The conclusions and recommendations of this thesis are presented in chapter V.

II. LITERATURE REVIEW

The literature review for this research was concentrated in two areas: recent thesis efforts which used simulation to study airlift problems, and software currently available for deployment planning.

THESES

Several recent theses have examined various aspects of airlift planning.

Hill and Donnelly studied the limitations to deploying C-130s from the CONUS to Europe during a major contingency. They built a simulation model to

> . . . analyze the interactions between the intertheater airlift flow and the deploying C-130s as they transit enroute facilities along the transatlantic routes (7:4).

They also used integer programming techniques to develop improved routing plans.

The deployment scenario was modeled with a network of eleven airfields using SLAM (Simulation Language for Alternative Modeling) and FORTRAN. The simulation output was used to conduct an experiment with four factors: 1) time of first launch from home station, 2) interval between launches, 3) route of flight, and 4) number of fuel trucks at enroute bases. The response variables observed were closure time (time from initiation of deployment to arrival

of last deploying aircraft at its destination) for the C-130s and enroute departure reliability for the strategic airlifters (7:42-43).

Hill and Donnelly concluded that resource limitations at enroute bases would not be a constraint to the rapid deployment of C-130s. They further found that

> . . . closure time of the C-130s can be reduced with the use of optimized routing plans. Additionally, more rapid generation of the C-130s would appear to allow significantly reduced closure times (7:ii).

They observed interaction between time of first launch from home station and route of flight. This led them to suggest that a decrease of both closure time and enroute congestion might be achieved by sending some of the earlier generating squadrons via longer but less congested routes, leaving the shorter routes for the later squadrons.

In addition to his thesis work with Donnelly, Hill modeled the deployment of C-130s to Europe using network programming software on a Z-248 computer. By looking at thirty-minute time periods, he was able to model a maximum of ten hours of the deployment before exceeding the software's capabilities. Hill concluded that a simulation using network programming techniques should be done on a minicomputer (6).

Cuda examined ramp space congestion during airlift operations at a single APOD during deployment and

transhipment of forces (4). APOD operations were modeled using SLAM, with C-5 and C-141 aircraft delivering a stream of cargo to be picked up by C-130s for forward delivery. Daily cargo delivery rates were set at 500, 1000, and 1500 tons, with the number of strategic and tactical aircraft unconstrained. Parking levels were calculated deterministically using standard onload and offload ground times.

By allocating parking space to tactical aircraft, tactical operations were unaffected over all combinations of arrival rate and parking space.

Cuda noted that while there was a strong relationship between strategic aircraft parking utilization and cargo diversions (as high as fifty percent cargo diverted), the diversion of C-130 aircraft did not begin until the parking utilization rate reached sixty-five percent (4:99-101).

Bowers used a SLAM model to analyze the performance of the Alaskan airlift system during wartime conditions. He found that the

> . . . number of aircraft and number of aircrews most directly affect system performance under 'average' operating conditions. To a lesser extent, length of crew day, season, and weather can exert significant influences on system operation. Factors such as physical limitations at airfields and materials handling equipment were found to have relatively small effects . . . (4:115)

He discovered the optimal ratio of aircrews to aircraft to be approximately 1.6, excluding aircrews utilized for ground duties, compared to a peacetime ratio of 1.5.

Software

Abreu and Pritchard developed the Force Closure Analysis Program (F-CAP), 'a tool for operational planners that automates the determination of the air transportation requirement and the closure time for a specified unit (1:i)'. F-CAP consists of two interactive programs useable on IBM PC or compatible computers. The Force Closure Simulation considers port constraints and airland/airdrop operations in verifying the feasibility of operational plans involving multiple APOEs and APODs. The Lift Asset Estimator calculates the number of airlift aircraft needed and computes tradeoffs between different types of aircraft. Like MAC's FLOGEN program, F-CAP is driven by cargo movement requirements. This makes F-CAP unuseable for planning beddown deployments.

The MAC Planner's Tool Kit (MPT) was developed by Maj. Brian Jones (8). MPT emulates the MAC FLOGEN program by reading the time phased force deployment requirements and accessing a database of 4400 airfields, producing a mission schedule. Written in Foxbase with Microsoft QuickBASIC subroutines, MPT may be used on a Z-248 or compatible

microcomputer. MPT produces a single mission schedule at a time, requiring manual deconfliction of missions.

Flogen (not MAC's FLOGEN) was written to automate the flow planning of a MAC strategic airlift Operational Readiness Inspection (ORI) scenario involving a single onload base, an offload base, and return to home station (3). A mission schedule and mission flow chart are produced. Flogen was written in BASIC for use on the Z-100 microcomputer using the CPM operating system.

Fragbuster was also written for the planning of strategic airlift ORIs, using as many as five different types of aircraft and transiting up to four bases (5). Output includes a mission schedule, mission flow chart, air operation order, and messages capable of AUTODIN transmission to the MAC AIMS database. Fragbuster was written in BASIC for use on the Z-100 and Z-150 microcomputers.

SUMMARY

The literature reviewed addressed several problems of both strategic and tactical airlift of cargo and troops. The only research dealing specifically with the deployment of C-130 aircraft to beddown locations in Europe was Hill's work with NETSID.

III. METHODOLOGY

OVERVIEW

The ideal CONUS-Europe C-130 deployment scenario would allow all aircraft to depart individual home bases and fly directly to their separate beddown locations, each aircraft proceeding via a different route. The elapsed time from initiation of the deployment to force closure (arrival of last deploying aircraft at its destination) would equal the flight time of the aircraft whose home station was the greatest distance from its beddown location.

The real world places several restrictions upon this scenario. Aircraft are based together and each base has limited maintenance, refueling, taxiway, runway and air traffic control resources. This forces some interval of time between departures. The factors which cause the sharing of home bases also encourage the sharing of beddown bases, requiring an interval between arrivals.

The distance between home base and destination requires at least one enroute stop for fuel. If each home base / beddown base pair had an exclusive enroute fuel stop, the home base departure interval would provide spacing for arrivals at the fuel stop. However, the limited number of airfields that can serve as enroute refueling stops dictates that these be shared by aircraft from different CONUS bases.

Arrivals at these shared airfields must be sequenced to avoid a traffic jam and the resulting slowdown of deploying aircraft.

Only three airfields are geographically positioned to serve as refueling stops in a CONUS - Europe deployment: Goose Bay in Labrador, Lajes in the Azores, and Keflavik in Iceland. Keflavik and Lajes are far enough from CONUS C-130 bases that most aircraft using these bases must make an additional refueling stop at a base in the eastern U.S. or Canada. These enroute refueling bases become a critical factor in a deployment of C-130s to Europe. Because of the volume of traffic transiting these bases, they become potential choke points in the deployment flow.

Deployment time can be minimized by using all available refueling bases and distributing the deploying aircraft among them as equally as operational considerations will allow. Each set of deployment routings which share a common enroute refueling stop could be planned as a separate deployment 'package', with the refueling airfield serving as the regulating parameter for that package.

Because the choke point is the only enroute stop common to all aircraft in a deployment package, the flow of aircraft must be regulated at that base. C-130s converging on the choke point will be flying at approximately the same airspeed and will spend approximately the same time on the

ground at enroute stops. If aircraft make stops at one or more intermediate airfields before and/or after the choke point, they will be sequenced through those bases in the same order and with the same minimum spacing as at the choke point.

An effective measure of an enroute airfield's ability to handle aircraft is the sustained rate at which transient aircraft can be launched. This launch rate is a function of the number of runways (usually one), the amount of parking space allocated to transient aircraft, maintenance and refueling capability, how broken aircraft will be repaired and resequenced into the flow, and the command, control and communications network supporting the coordination of all resources. The launch rate will be inversely proportional to the minimum interval between aircraft departures. A short interval would indicate efficient operation at the choke point and reduced closure time for forces transiting that base. The interval between departing aircraft at the choke point would be a good control parameter for the deployment flow.

ASSUMPTIONS AND LIMITATIONS

The following assumptions were made :

1. For the specific deployment contingency, the route of flight from home station to beddown location has been determined for each CONUS C-130 squadron.

2. The planner has determined choke points for various deployment routes by analyzing historical deployment exercise data and surveys of airfield capabilities.

3. Stage crews will be prepositioned at enroute airfields before deploying aircraft arrive, allowing aircraft to continue movement toward the beddown locations without waiting for crews to rest.

4. MAC has surveyed airfields worldwide and assessed the aircraft handling capacity at each airfield for two categories of aircraft: wide-body (C-5, KC-10, DC-10, Boeing 747) and narrow-body (C-141, C-130, C-135, C-9, Boeing 707 and 727, McDonnell-Douglas DC-8 and DC-9 and smaller aircraft). This airfield capacity is a function of ramp space, refueling capability, fire fighting resources, and all other characteristics of the airfield which affect air operations. In a contingency or exercise a portion of the total airfield capacity will be allocated to the deploying aircraft. The remainder would be reserved for deploying fighters or tankers. For this thesis MOG (Maximum On

Ground) is defined as the portion of that airfield's total aircraft handling capacity which has been allocated to MAC for the deployment. A separate MOG will be allocated for narrow and wide aircraft.

5. The planner has determined the CONTROL INTERVAL, the time between aircraft departing the choke point. Some aircraft arriving at the choke point will require maintenance that cannot be completed during their planned ground time. The control interval must include enough slack time to reinsert these aircraft into the traffic flow without disrupting the deployment schedule.

6. To preserve the unclassified nature of this thesis, beddown airfields used in all runs of DEPLOY were chosen without knowledge of actual plans. Home stations of USAF C-130 squadrons are unclassified and published each year in Air Force Magazine.

SOLUTION TECHNIQUE

DEPLOY, an interactive program, was written based on the concept of using departure times from the choke point as the mechanism for regulating the deployment flow. The choke point takeoff time of each deploying aircraft is the aircraft's CONTROL TIME. The planner enters the date and time at which the choke point can begin a sustained aircraft launch rate and the control interval between aircraft

departures from the choke point. A series of menus leads the planner through the development of the deployment flow plan.

Arrival time at the choke point for the first deploying aircraft is calculated by subtracting that aircraft's choke point ground time from the control time. Flight and ground times for legs into the choke point are subtracted from this arrival time to develop a schedule for the first part of the mission. The schedule is completed by adding flight and ground times for legs flown after the choke point to the control time. The control time for each aircraft is determined by adding the control interval to the control time of the previous aircraft.

As each aircraft's schedule is built, the actual number of aircraft on the ground at the choke point is checked. If an arriving aircraft would exceed MOG, its arrival at the choke point is delayed until a slot is available. This delay will be reflected in that aircraft's arrivals at and departures from all bases transited prior to the choke point.

BASIC was chosen in order to make it possible to incorporate the program into a flow planning package which HQ MAC/XOSP plans to develop (8). A program user's manual is presented

in Appendix A and a listing of the program is found in Appendix C. Appendix B contains a list of variables and diagrams of the arrays used in DEPLOY.

PROGRAM DESCRIPTION

The program begins by reading two data files, SQUAD#.DAT and BASE#.DAT. SQUAD#.DAT contains records for each of thirty-nine active duty, Air Force Reserve and Air National Guard squadrons stationed in the CONUS (12). Each record contains six fields:

1 - Squadron identifier

2 - Type, i.e. USAF, Reserve or Air National Guard

3 - Type of aircraft with which the unit is equipped (C-130A/B/E or C-130H)

4 - Number of aircraft assigned

5 - Home station

6 - Available Date (day after being alerted that squadron will be ready to deploy).

BASE#.DAT contains records for seventy airfields. Each record has the following fields:

1 - Name of airfield

2 - ICAO (International Civil Aeronautical Organization) identifier

3 - Latitude

4 - Longitude

5 - MOG for narrow-bodied aircraft

6 - MOG for wide-bodied aircraft

The remainder of the program is menu-driven, requiring the planner to make inputs with the keyboard. A menu is displayed, offering the planner six options; build a new deployment plan, change the plan currently in the computer's memory, retrieve a plan from disk storage, save a plan on a disk, process the plan (to produce a schedule, closure and fuel requirements), add a squadron to (or change data in) the SQUAD#.DAT data file, and add an airfield to the BASE#.DAT data file.

To build a deployment flow plan the planner must enter types of aircraft, number of deploying aircraft from each squadron, enroute ground times, routes of flight, average wind effect (wind factor) along each route and the first departure time from the choke point.

Characteristics of five types of aircraft (C-130A/B/E, C-130H, C-141, C-5, KC-10) have been coded into the program and the planner is given the option of adding up to four more. Aircraft parameters are aircraft identifier, body type (wide or narrow),and block speed (average speed for a leg). Because block speed is based on time to accelerate to cruise speed, the total time spent at cruise speed, and time to decelerate for approach and landing, block speed will vary as a function of leg length. The block speeds coded

into DEPLOY are based on leg length of three thousand miles. The planner is able to change ground times from those in AFP 76-2 if desired.

The planner enters the route of flight for each deploying unit. Although most C-130s deploying from the CONUS to beddown bases in Europe will require no more than two refueling stops, as many as four enroute stops may be planned, allowing a great deal of flexibility. For each flight leg the planner enters the average headwind/tailwind in nautical miles per hour.

Wind factor (effective headwind or tailwind), is entered for each leg to be flown. This is added to the aircraft's block speed to get ground speed. The great circle length of each flight leg is then computed using the following equations:

$$D = \arccos[(\sin L1 \sin L2)+(\cos L1 \cos L2 \cos DLo)] \quad (1)$$

where

L1 = latitude of point of departure in radians

L2 = latitude of destination in radians

DLo = difference of longitude in radians of the points of departure and destination (2:1258)

Distance in nautical miles = (3439.77)(D) (2)

where 3439.77 is the mean radius of the earth in nautical miles (2:1117). The distance is divided by ground speed to determine the time of flight for each leg.

Mission 'templates' must be built by matching each deploying unit with a route of flight. The mission template is used to assign a ground time to each enroute stop based on whether the aircraft will onload or offload cargo, perform an engine-running offload, or refuel and depart. These ground times are in accordance with Air Force Pamphlet 76-2. A mission number may be assigned to each mission template.

The planner designates which airfield will be the choke point and the time and date at which the choke point can begin supporting the deployment flow. The planner may also change MOG at the choke point. A reduction in MOG would be necessary if maintenance or refueling capability were reduced because of an accident or enemy action. MOG might be increased if more of the airfield's total capacity were allocated to the deployment.

From the processing menu the planner selects the squadrons to deploy and the number of aircraft from each squadron. A squadron having sixteen C-130s may be deployed as a single unit or broken into smaller units of one or more

aircraft. The order of flow through the choke point will be determined by each squadron's Available Date. Units having the same Available Date will flow in the order they were selected by the planner.

A sequencing algorithm separates the deploying aircraft at the choke point by inserting the control interval between choke point departures. The resulting deployment schedule is then adjusted to insure that MOG at the choke point has not been exceeded. If the number of C-130s on the ground is greater than the MOG for narrow aircraft, arriving C-130s can use unfilled slots for wide aircraft (wide aircraft cannot use narrow slots). In the event all slots are filled, the next inbound aircraft will be scheduled to arrive when a slot is given up by a departing aircraft.

The resulting schedule may be tailored to a specific contingency by using four utility program modules provided. The utilities allow an individual mission or a series of missions to be moved to a time earlier or later than scheduled, flight and ground times to be changed, and missions to be repositioned in the flow plan.

The program provides three types of output; a schedule, a closure summary and a fuel requirement summary. The schedule may be obtained for all or any part of the deploying force. It can be viewed on the screen or ported to a printer. The closure summary is a list of deploying

squadrons or parts of squadrons in order of flow through the choke point. Date and time are given for the departure of the first aircraft from home station and the arrival of the last aircraft at beddown base for each deploying unit. The fuel requirement summary shows fuel for the deployment that will be required at each origin and enroute airfield.

A more detailed description of the program is found in the User's Manual in Appendix A.

IV. RESULTS

To demonstrate a potential use of DEPLOY, an experiment was conducted to compare closure times of various deployment scenarios. In the experiment, each active duty CONUS squadron deployed sixteen aircraft to a fictitious beddown location in Europe. Goose Bay Air Base, Labrador was used as the enroute refueling stop, or choke point.

EXPERIMENTAL DESIGN

The experimental factors chosen were choke point departure interval (INT) and sequencing of squadrons (SEQ). INT was tried at fifteen, thirty and forty-five minutes. SEQ was varied at three levels: squadrons closest to the choke point deploying first (CLS), squadrons furthest from the choke point deploying first (FAR), and alternating close and distant squadrons (ALT). The experimental design is shown in Table 4-1.

Table 4-1 TREATMENT COMBINATIONS

			SEQ		
			CLS	ALT	FAR
			1	0	-1
	15 min	1	1, 1	1, 0	1,-1
INT	30 min	0	0, 1	0, 0	0,-1
	45 min	-1	-1, 1	-1, 0	-1,-1

The results of the experiment are shown in Table 4-2.

Table 4-2 GOOSE BAY RAW DATA

(minutes to close)

		SEQ		
		CLS	ALT	FAR
	15 min	2800	2810	2890
INT	30 min	4705	4715	4795
	45 min	6610	6620	6700

SAS (12) was used to plot the dependent variable CLOSURE TIME as a function of INT and SEQ. The response surface which resulted is shown in Figure 4-1.

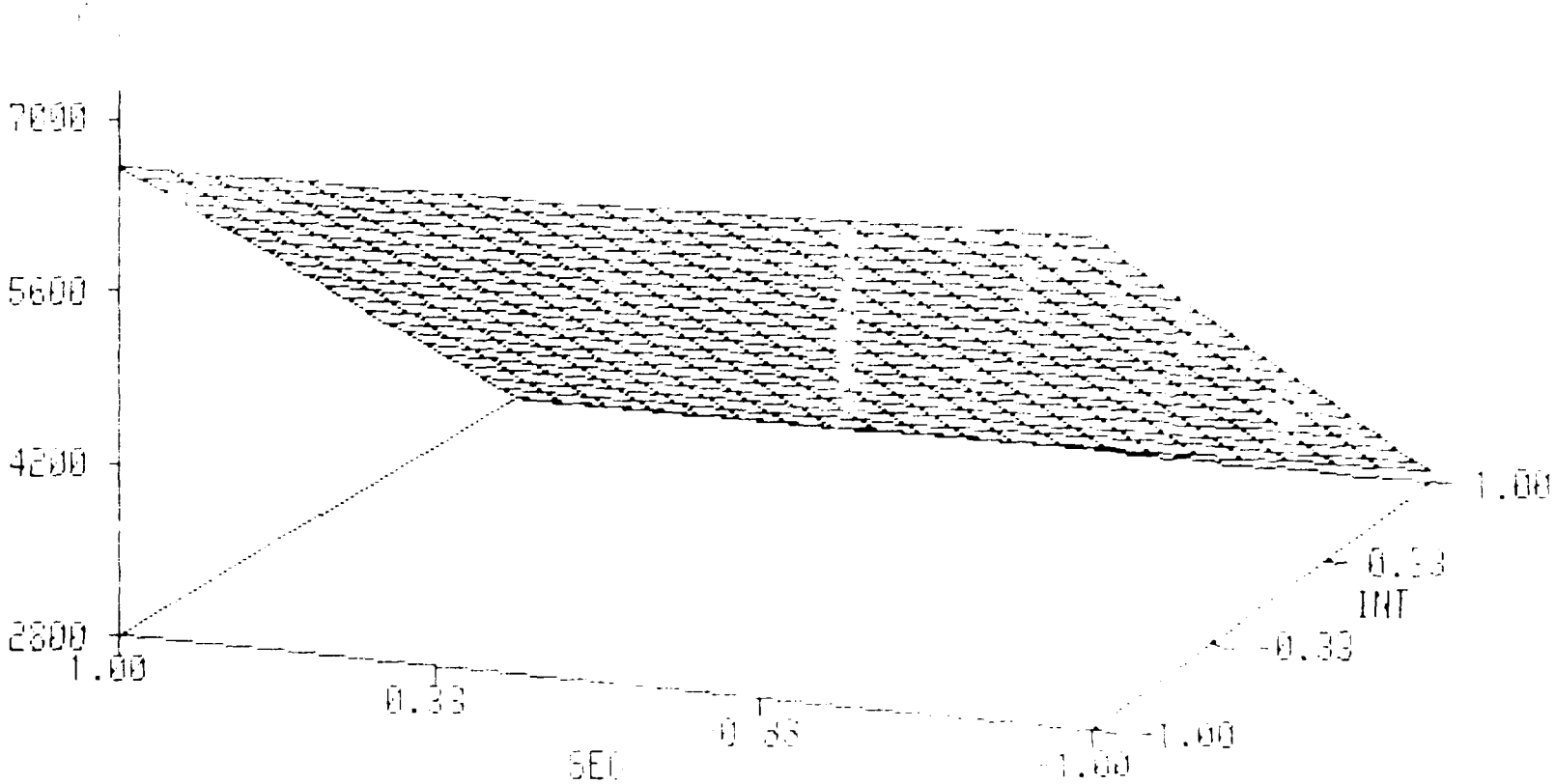


Figure 4-1 Closure For Goose Bay

The experiment was repeated with Lajes Air Base, Azores, as the choke point. The results of that experiment are shown in the Table 4-3 and on the response surface in Figure 4-2.

Table 4-3 LAJES RAW DATA

(minutes to close)

		SEQ		
		CLS	ALT	FAR
INT	15 min	2860	3045	3120
	30 min	4765	4945	5025
	45 min	6670	6855	6930

The response surfaces indicate that closure time is very sensitive to changes in choke point departure interval. This is to be expected since each minute added to the interval would increase closure time by [the number of deploying aircraft - 1] minutes.

An unexpected inference from the response surfaces is closure's relative lack of sensitivity to the sequencing of squadrons through the choke point. A C-130 flying from Dyess AFB, Texas, to Goose Bay, Labrador, would have a no-wind flying time of seven hours and twenty-five minutes,

LAJES

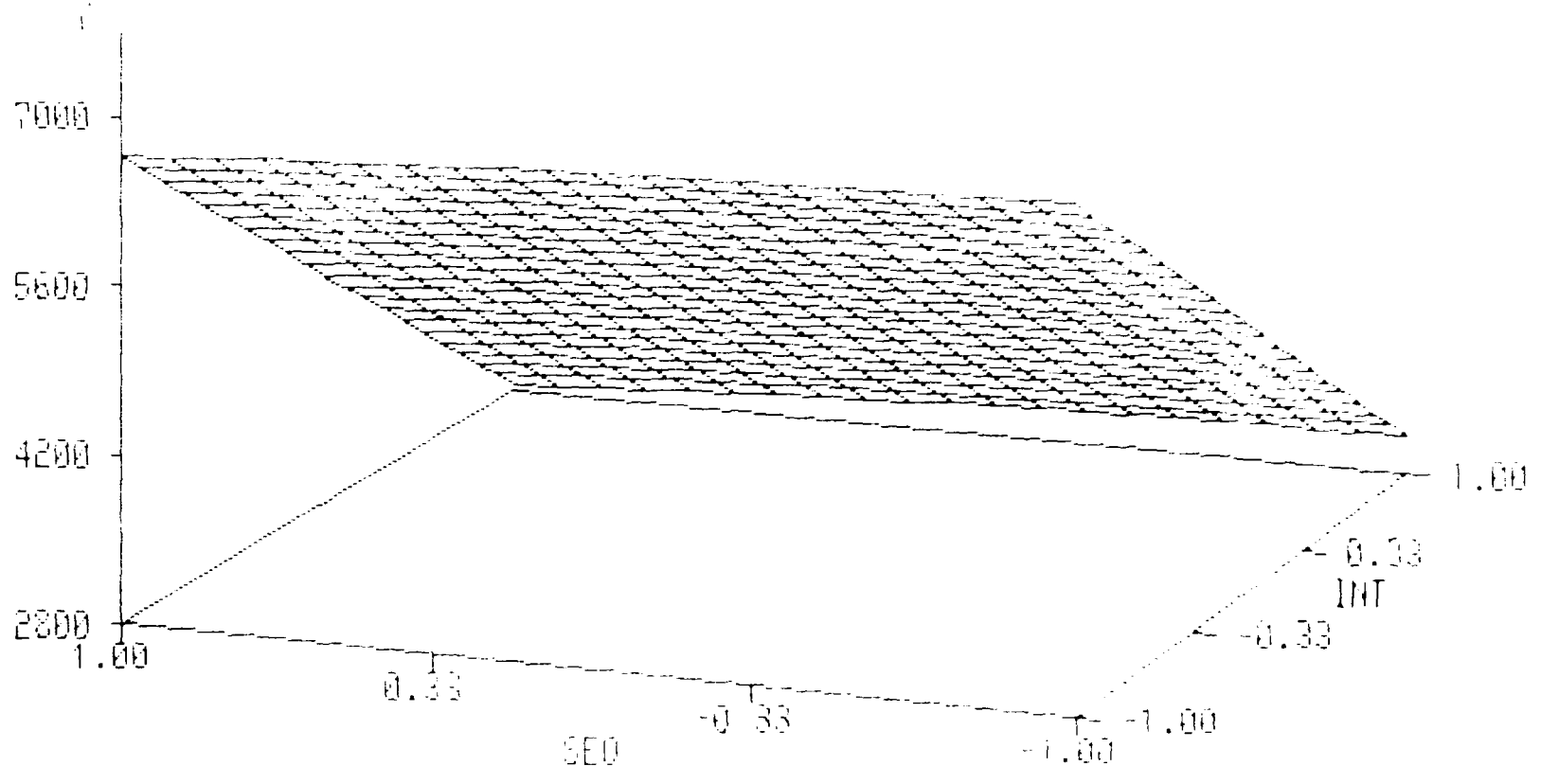


Figure 4-2 Closure For Lajes

while a C-130 departing Pope AFB, North Carolina, for Goose Bay would fly for four hours and forty-five minutes. If a squadron stationed at Pope AFB were to deploy immediately after a Dyess squadron, the first Pope aircraft would depart Pope after the last Dyess aircraft had been airborne for two hours and forty minutes (7+25 - 4+45), plus the Goose Bay departure interval. If the sequence were reversed, the first Dyess C-130 could take off for Goose Bay two hours and forty minutes (minus the choke point departure interval) BEFORE the last aircraft departed Pope. While this makes it appear as if a sizeable reduction in closure time would result from scheduling the squadrons in order of their proximity to the choke point, the reduction is small when compared to the total time required to close the deployment.

The response surface for a deployment via Lajes shows the same sensitivity to choke point departure interval as the deployment via Goose Bay. A slightly increased sensitivity to squadron sequencing is due to the longer time that C-130s must fly in order to reach Lajes.

In both experiments MOG was intentionally set at a large value so it would not be a limiting factor. Limiting MOG can be determined by the equation:

$$MOG = GT / INT \qquad (3)$$

where

GT = planned ground time

INT = planned departure interval

As long as the MOG is greater than or equal to (GT/INT) the aircraft will flow smoothly, one departing every INT minutes. When MOG is less than (GT/INT) the aircraft will flow from the choke point in surges. During the surges aircraft will depart the choke point every INT minutes. Between the surge periods the choke point will be operating at a slower pace, with aircraft departing at intervals greater than INT minutes. When MOG is exactly (GT/INT) the choke point will operate at *continous* rate with *no* departure surges and all aircraft departing at INT minute intervals.

Both experiments were repeated with the MOG at each choke point set equal to two. At departure intervals of fifteen and thirty minutes, flow surges occured since:

$$GT / INT = 90 / 15 = 6 > MOG \quad (4)$$

for INT = 15 minutes and

$$GT / INT = 90 / 30 = 3 > MOG \quad (5)$$

for INT = 30 minutes.

During these surges, aircraft are departing the choke point at INT minute intervals.

At departure intervals of forty-five minutes there was a smooth flow of aircraft, due to:

$$GT / INT = 90 / 45 = 2 = MOG \tag{6}$$

The results of the experiments are shown in Tables 4-4 and 4-5 and Figures 4-3 and 4-4.

Table 4-4 GOOSE BAY, MOG = 2

(minutes to close)

		SEQ		
		CLS	ALT	FAR
INT	15 min	6520	6525	6600
	30 min	6535	6540	6615
	45 min	6550	6555	6645

Table 4-5 LAJES, MOG = 2

(minutes to close)

		SEQ		
		CLS	ALT	FAR
	15 min	6555	6600	6850
INT	30 min	6610	6615	6865
	45 min	6625	6630	6680

The response surfaces in Figures 4-3 and 4-4 are relatively flat and show an increased closure time when compared to the surfaces in which MOG was not a factor. This occurs because the slower periods between surges are adding to closure time without moving aircraft. The overall effect of MOG restricting the flow is the increase in closure time and the reduced sensitivity of the flow to choke point departure interval.

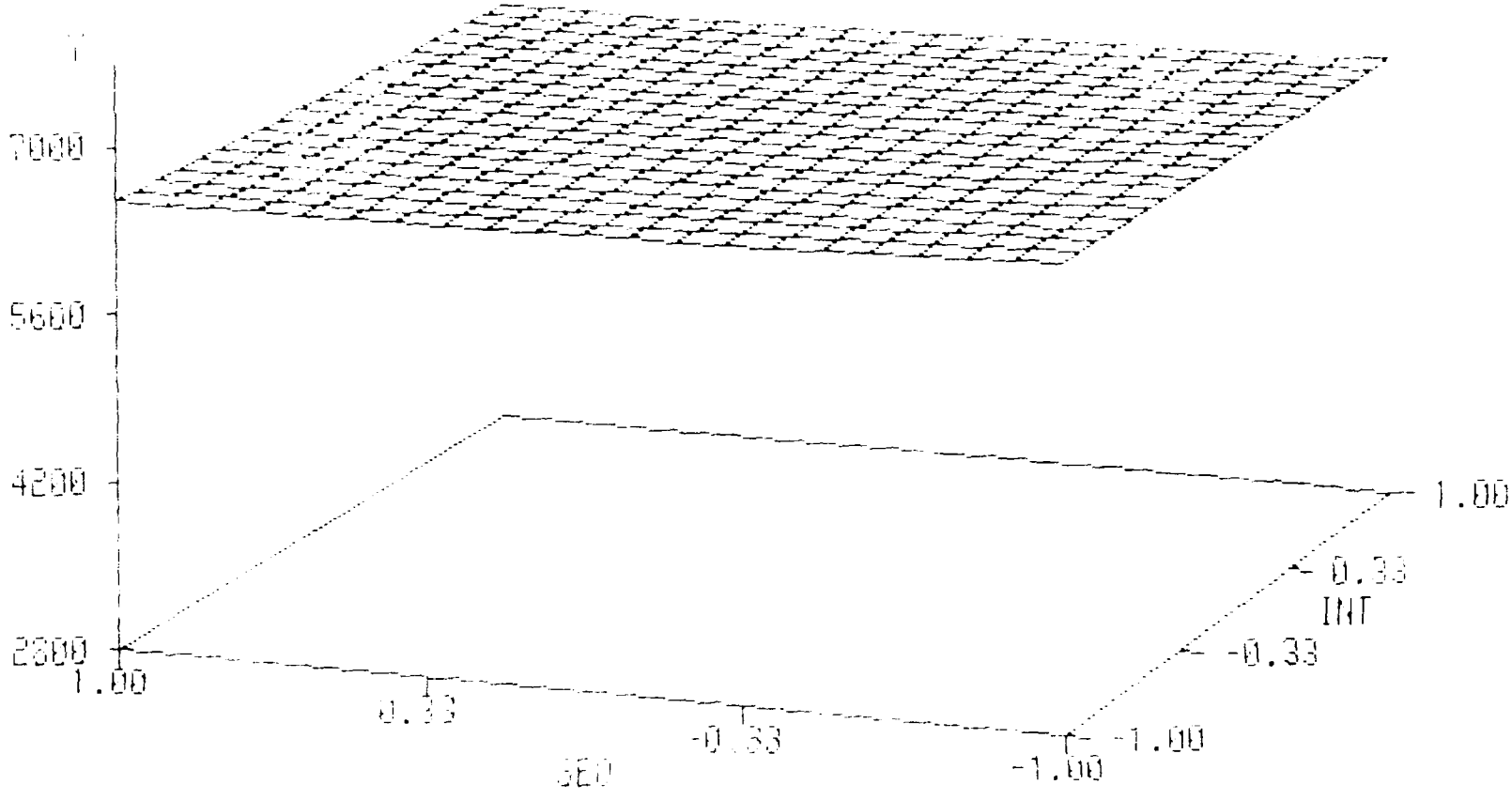


Figure 4-3 Closure For Goose Bay, MOG = 2

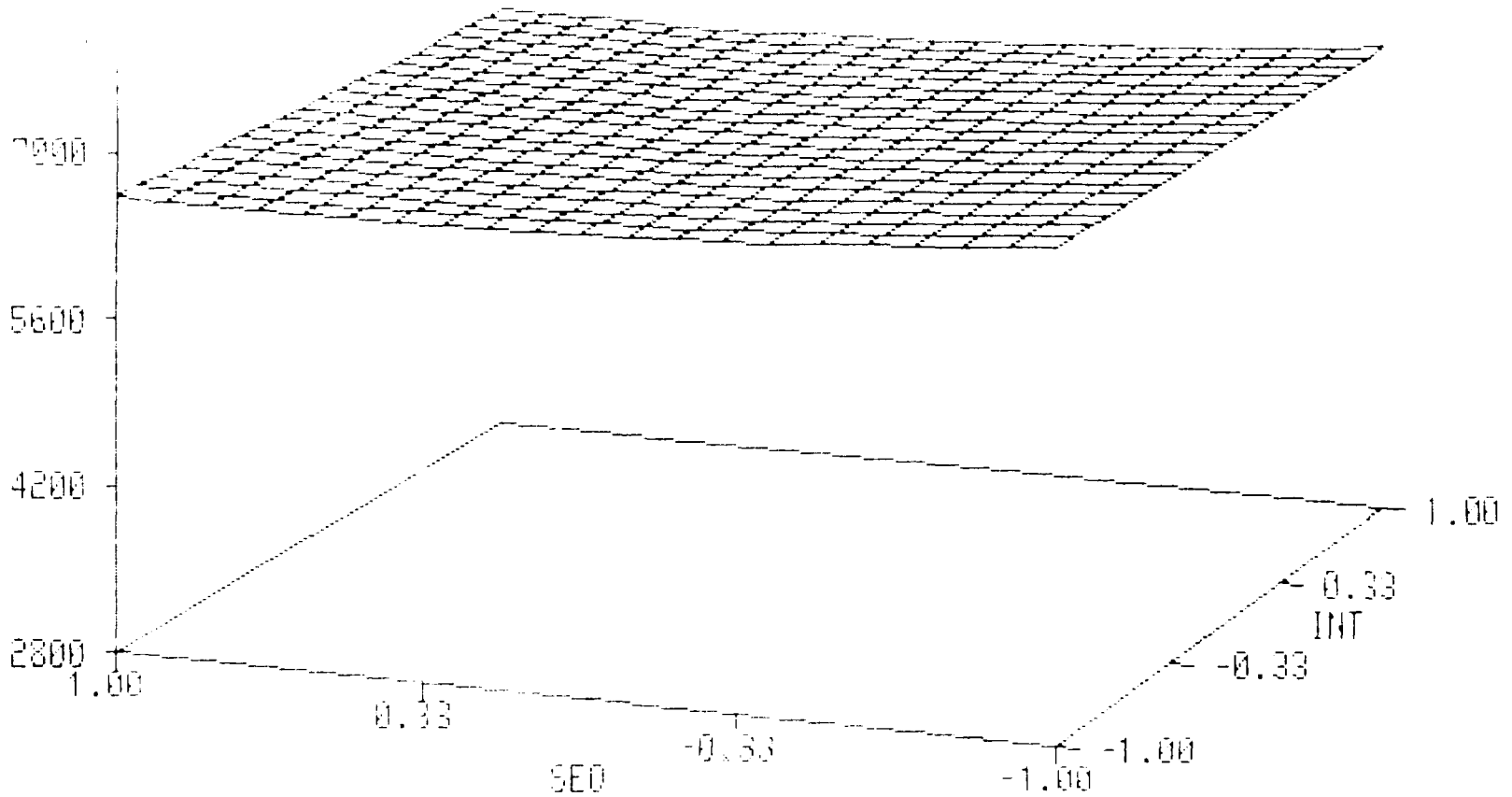


Figure 4-4 Closure For Lajes, MOG = 2

SUMMARY

While the experiments themselves are interesting, the usefulness of DEPLOY is demonstrated by the fact that the thirty-six C-130 deployments generated for the experimental data, each containing one hundred twenty-eight aircraft, were scheduled in less than three hours. Manually planning a single deployment of all CONUS-based active duty squadrons would take several hours. When this planning time is multiplied by the thirty-six deployments used in the experiment, the advantage of using an interactive tool like DEPLOY becomes apparent.

V. CONCLUSIONS AND RECOMMENDATIONS

The objective of this research was to develop a methodology for automating deployment planning of C-130 aircraft from CONUS bases to beddown locations. This was accomplished by means of an interactive program which runs on a Zenith Z-248 (IBM AT compatible) microcomputer. The DEPLOY program reduces the time required to plan a deployment of all CONUS-based C-130s from an estimated two days to less than two hours. Deployment flow plans may be saved on disk and retrieved for later use or modification. Output includes a schedule for each aircraft, closure for each deploying unit, and fuel required at each airfield involved in the deployment plan.

While DEPLOY is a marked improvement over the current manual method of deployment flow planning, improvements could be made to increase its usefulness. Suggestions for several improvements are noted below.

1. Add a program module to select the optimal routing for each squadron. Routings which were determined during peacetime deliberate planning sessions may prove unusable as the political and military situations change.

2. Add a module which determines the choke point, given the airfields used for enroute refueling.

3. Modify the wind factor subroutine to calculate wind factors from winds entered by the planner. This would save the planner the step of calculating the wind factors manually.

4. Add logic which would limit the flow of aircraft through an airfield to that which could be supported by fuel supplies at that airfield.

5. Add logic to prevent the planner from building a mission template with a route segment exceeding an aircraft's maximum range. This must currently be checked by the planner and could be overlooked.

6. The number of routes is currently limited to twenty, as is the number of mission templates. The program should be modified to allow the planner to set these limits to match his needs and the memory available in his computer. The use of dynamic arrays in QuickBASIC will make this an easy modification.

7. DEPLOY uses block speed based on a flight leg length of three thousand miles. A look-up table could be added to cover a range of leg lengths.

8. Investigate the possibility of modifying DEPLOY for planning fighter deployments. This is another beddown deployment problem which will occur at the onset of a crisis, when time is at a premium.

DEPLOY

USER'S MANUAL

DEPLOY is a tool for planning the deployment of aircraft from their home stations to their theater beddown locations. Although intended primarily for C-130s, the program also contains parameters for C-141, C-5 and KC-10 aircraft. It was written for use on a Z-248 microcomputer with 512 kilobytes of RAM and either a hard disk drive or one floppy disk drive. The source language is Microsoft QuickBASIC 4.0.

The program disk must contain four files.

- DEPLOY.EXE - the executable program

- BASRUN40.EXE - the BASIC runtime module

- BASE#.DAT - the airfield data file containing the name, ICAO identifier in capital letters, latitude and longitude in degrees and tenths of a degree, narrow and wide MOG of each airfield. Bases may be added using the ADD AIRFIELD TO DATABASE option of the MAIN MENU explained later in this manual or with a text editor such as edlin.com.

- SQUAD#.DAT - the squadron data file containing the squadron identifier, type (1=active, 2=Res, 3=ANG), type acft, number of acft, home base (ICAO), beddown base (ICAO),

day available (0, 1, 2, etc) and beddown deadline (day 1, 2, etc) for each squadron. Changes and additions may be made using the CHANGE SQUADRON DATA option of the MAIN MENU explained below.

The program is initiated by typing [DEPLOY] [return]. After the base and data files are read, the main menu will appear.

MAIN MENU

---

1 ..... CREATE A NEW FLOW PLAN

2 ..... CHANGE CURRENT FLOW PLAN

3 ..... LOAD FILE

4 ..... SAVE FILE

5 ..... CHANGE SQUADRON DATA

6 ..... ADD AIRFIELD TO DATABASE

7 ..... FLOW MENU

8 ..... END

ENTER REQUEST ?

1. CREATE A NEW FLOW PLAN

2. CHANGE CURRENT FLOW PLAN

Selection of either of the first two menu items will present the INPUT CHECKLIST. Item 1 clears memory, allowing a new

plan to be entered while Item 2 allows changes to be made to an existing flow plan.

3. LOAD FILE

Used to load a flow plan file previously saved on disk. * See note below.

4. SAVE FILE

Saves a flow plan file on disk. The file must be given a filename prefix in accordance with the MS-DOS convention. The .DAT suffix will be added by DEPLOY. A flow plan containing one hundred fifty missions will take approximately 100 kb of disk space. * See note below.

* note : Files may be saved to and loaded from any disk drive, however, only the files on the program (logged) drive will be shown on the screen.

5. CHANGE SQUADRON DATA

Up to five squadrons may be added to the SQUAD#.DAT data file each time DEPLOY is run. Five parameters are tracked for each squadron and may be changed in the CHANGE SQUADRON DATA routine :

- TYPE SQUADRON - active duty, Reserve or ANG.

- TYPE ACFT ASSIGNED - C-130A/B/E, C-130H, C-141, C-5, KC-10, any others added by the user in the ACFT TYPES option of the INPUT CHECKLIST.

- PAA - the number of airframes assigned to the squadron.

- HOME BASE - the squadron's home station

- AVAILABLE - the day the squadron will be available for deployment (0, 1, 2, etc; day 0 being the day the planner is alerted to flow plan).

6. ADD AIRFIELD TO DATABASE

Each time DEPLOY is run, five airfields may be added to the BASE#.DAT database. Six characteristics of each airfield are tracked :

- ICAO identifier

- Name of airfield

- Latitude in degrees and tenths of a degree. Northern latitude is entered as a positive number, Southern latitude as a negative number.

- Longitude in degrees and tenths. Enter Western longitude as a positive number, Eastern longitude as negative.

- Narrow MOG

- Wide MOG

DEPLOY may be terminated by selecting the END option of the MAIN MENU. The opportunity is offered to save any changes that were made to the squadron or airfield data.

INPUT CHECKLIST (completed (**))

() 1 ..... AIRCRAFT TYPES

() 2 ..... GROUND TIMES

() 3 ..... FLIGHT ROUTINGS

() 4 ..... ROUTE WIND FACTORS

() 5 ..... MISSION TEMPLATES

() 6 ..... CONTROL BASE, TIME, DATE

7 ..... REVIEW AIRFIELDS

8 ..... CHECK FLYING TIMES

9 ..... RETURN TO MAIN MENU

ENTER REQUEST ?

1 - AIRCRAFT TYPES

Allows up to four additional types of aircraft to be added. Aircraft designator, body type (wide/narrow), block speed

and average fuel consumed (1000s of pounds/hour) are the required entries. The screen below appears when item 1 is selected:

PROGRAM CONTAINS PARAMETERS FOR THESE AIRCRAFT

C-130A/B/E	C-130H	C-141	C-5	KC-10

'ENTER' TO CONTINUE, 'A' TO ADD AIRCRAFT ?

2 - GROUND TIMES

Allows review or modification of the aircraft ground times (engine running offload, upload, download, enroute stop) specified in AFP 76-2. Called automatically if an aircraft type was added. Option 2 gives the following screen:

	ERO	DNLOAD	UPLOAD	EN ROUTE
1 C130	00+30	01+30	01+30	01+30
2 C130H	00+30	01+30	01+30	01+30
3 C141	00+45	02+15	02+15	02+15
4 C5	01+00	03+15	03+45	02+15
5 KC10	00+00	03+00	05+00	01+30

'C' TO CHANGE ONE OR MORE GROUND TIMES,
'ENTER' TO RETURN TO MENU ?

3 - FLIGHT ROUTINGS

Used to enter the route of flight for deployment. Twenty routes are permitted, each having up to five legs. ICAO

identifiers for each base on the route are entered, IN CAPITAL LETTERS. Bases not in the BASE#.DAT file will be rejected. Selecting option 3 presents the screen below:

```
    ENTER ICAOS OF ROUTE OF FLIGHT (RETURN AFTER EACH ICAO)
1   KPOB    LPLA    EGUN
2
.
.
.
19
20
  ENTER THE NUMBER OF ROUTING TO CHANGE (ENTER TO END)  ?
```

4 - ROUTE WIND FACTORS

Used to enter or change the wind factor (effective tail wind) in knots for each leg entered in FLIGHT ROUTINGS. Head winds are entered as negative numbers. The distance in nautical miles between the departure and arrival base of each leg is computed from the latitude and longitude of the bases. The distance and wind factor of each leg are saved and used with the block true airspeed of each type aircraft to compute leg flying time. Called automatically if one or more flight routes were entered. The wind factors (in knots) are changed on this screen:

```
          FROM        TO       WF

      1   KPOB  TO  LPLA  =  15
      2   LPLA  TO  EGUN  =  ?

ENTER NEW WIND FACTOR FOR THIS LEG. 'ENTER' IF NO CHANGE
```

5 - MISSION TEMPLATES

A squadron is entered and matched with a routing. You are given the option of a quick turn at each enroute stop or designating an upload, download, or engine running offload (ERO). The ground times will be those entered or accepted in the GROUND TIMES option described above. A seven-digit mission number may be entered. The takeoff sequence (by base of origin) and Julian day of mission origin are automatically added to enable identification of individual missions. Twenty mission templates are permitted.

```
                        MISSION PLANS

1 C130   KPOB   LPLA   EGUN              40TAS *QKTRN*
2 C130   KPOB   KLIZ   LPLA   EGUN       39TAS *QKTRN*
.
.
.
19
20
  ENTER TEMPLATE TO CHANGE.  HIT ENTER TO END  ?
```

6 - CHOKE POINT, TIME, DATE

The ICAO identifier of the choke point is entered. The MOG for narrow and wide aircraft is printed on the screen with the option of changing them. When DEPLOY is ended there will be an opportunity to save these changes in the BASE$.DAT file. The desired time and date of the first

takeoff from the choke point are entered, followed by the interval between takeoffs from the choke point, as shown below:

```
        DEPARTURES ARE REGULATED FROM THE CHOKE POINT

ENTER ICAO OF CHOKE POINT                              ? LPLA
NARROW MOG  16  'ENTER' IF OK, 'C' TO CHANGE           ?
WIDE MOG     2  'ENTER' IF OK, 'C' TO CHANGE           ?
ENTER FIRST TAKEOFF TIME FROM CYYR (HHMM)              ? 1000
ENTER DATE  (DDMMYY)  OF T.O.                          ? 180388
ENTER INTERVAL BETWEEN CYYR TAKEOFFS (IN MINUTES)      ? 45
```

7 - REVIEW AIRFIELDS

Allows viewing of all airfields and ICAO identifiers in the BASE# DAT file. This is provided as a memory aid and a means to print a listing (shift-Prt Sc).

8 - CHECK FLYING TIMES

Used to view flying times of all routes for each aircraft type. If it appears that realistic times have been exceeded, the routing may be changed in the FLIGHT ROUTINGS routine. Called automatically if a routing has been entered

in FLIGHT ROUTINGS or if WIND FACTORS were entered/changed. Flight times are presented for each leg in the format shown below for C130s:

```
-> C130 <-

KPOB  9+10  LPLA  5+15  EGUN
KPOB  3+15  KLIZ  6+50  LPLA  5+15
```

As each item on the INPUT CHECKLIST is selected, the checklist box () to the left of the item will be filled (**), whether or not any data is entered. In addition, the () boxes of all affected items will be cleared. This only clears the checklist () boxes, not the data entered when those checklist items were selected. If item 9 - RETURN TO MAIN MENU - is selected while any checklist boxes are empty, an 'INCOMPLETE ---->' warning will be printed to the left of the () box. A second attempt to return to MAIN MENU will be successful, as the program assumes you have selected that option intentionally.

If the INPUT CHECKLIST is entered via the CHANGE CURRENT FLOW PLAN option, all checklist boxes will be filled (**) when the checklist first appears.

After the INPUT CHECKLIST has been completed. The user should return to the MAIN MENU and save the 'raw' data. This will save time later on if a way is found to bypass the 'foolproofing' mechanisms and crash the program.

After saving the data select item 7, FLOW MENU. If the INPUT CHECKLIST was not completed, a message will appear, directing either a return to the INPUT MENU or the loading of a flow plan from disk.

FLOW MENU

---

1 ..... FLOW AND MOG

2 ..... SLIDE MISSIONS

3 ..... CHANGE MISSIONS

---

4 ..... MISSION SCHEDULE

5 ..... FUEL REQUIRED

6 ..... FORCE CLOSURE

---

7 ..... MAIN MENU

ENTER CHOICE ?

The FLOW MENU is divided into two sections. The first three options build the flow plan.

1 - FLOW AND MOG

Missions are selected from the mission templates. As a template is selected, the total number of aircraft available to deploy is displayed on the screen. As each desired template is entered, the squadron's available aircraft are depleted by the number deployed. The aircraft will flow

through the choke point in the order the templates are selected, with the following exception: squadrons will be sorted by available date. Two squadrons having the same available date will flow in the order selected.

When the last desired template has been entered and 'ENTER' selected, DEPLOY will sort the templates by available date and break them into individual missions. These missions will depart the choke point at the interval entered in the INPUT CHECKLIST (item 6).

If the number of aircraft on the ground at the choke point exceeds MOG for that body type (wide/narrow), further arrivals will be delayed until an aircraft departs. Wide-body aircraft may use only wide slots. Narrow aircraft may use wide slots once they have filled all narrow slots.

2 - SLIDE MISSIONS

Allows an individual mission or a series of missions to move to an earlier or later time. Two cautions must be observed when sliding missions: 1) beware of moving them 'on top' of other missions. 2) move missions no more than twenty-one days from the original scheduled time.

3 - CHANGE MISSIONS

This option presents the CHANGES MISSIONS menu.

CHANGE MISSIONS

1 ..... MISSION NUMBERS

2 ..... CHANGE GROUND TIMES (REFLOW)

3 ..... CHANGE FLIGHT TIMES (REFLOW)

4 ..... REPOSITION IN THE FLOW (REFLOW)

5 ..... LINK / UNLINK MISSIONS

6 ..... RETURN TO FLOW MENU

ENTER CHOICE ?

› 1 MISSION NUMBERS enables you to change the seven-character prefix or the eighth and ninth 'sequence' digits of the mission number.

› 2 CHANGE GROUND TIMES allows the ground time at any enroute base to be changed from the time specified in the INPUT CHECKLIST.

› 3 CHANGE FLIGHT TIMES allows the flight time of any leg to be changed from that computed by DEPLOY.

› 4 REPOSITION IN THE FLOW lets a mission be moved anywhere in the flow. Also allows two missions to be swapped.

› 5 LINK / UNLINK MISSIONS permits a mission which terminates at a base to be linked to a mission originating at that base, provided the same type aircraft is used on both missions.

Options four through six are the output options.

4 - MISSION SCHEDULE

Prints a schedule of all or part of the flow plan to the screen or printer. Data on individual missions is printed in the format shown below.

```
FLOW SEQUENCE #    1      CYYR DEPARTURE TIME  15 JAN 1200
ACFT C130       39  TAS        MISSION #  PEN0602010015

ARRIVE          GNDTME              DEPART          FLTTM     LEG
                              KPOB  15 JAN 0545     4+45      1
15 JAN 1030     1+30          CYYR  15 JAN 1200     7+30      2
15 JAN 1930                   EGUN
```

The two-line schedule header shows that this is the first mission in the flow (sequence # 1) and departs the choke point (CYYR) at 1200 on 15 JAN. The mission will be flown by a C-130(A, B, or E) from the 39th TAS as mission number PEN0602010015. The eighth and ninth characters of the mission number (01) indicate this is the first departure from the base under the mission prefix PEN0602. The mission originates on the 015 Julian day, as shown by the last three characters of the mission number.

The body of the schedule gives the planned itinerary and is read as follows:

Leg 1: Depart Pope AFB at 0545 on 15 JAN, arrive Goose Bay at 1030 on 15 JAN after a 4+45 flight. Ground time at Goose Bay is 1+30.

Leg 2: Depart Goose Bay at 1200 on 15 JAN, arrive Mildenhall 1930 15 Jan after flying for 7+30.

No ground time is specified at Mildenhall since the mission terminates there.

The leg numbers in the right-hand column are referenced when making changes to flight and ground times as described previously under the CHANGE MISSIONS option.

5 - FUEL REQUIRED

Depicts the fuel required by the flow plan at each base and total fuel required, in one thousand pound units. The total number of flying hours for the plan is also given. This information is printed only on the screen. A hard copy may be made by typing [shift-Prt Sc].

6 - FORCE CLOSURE

Shows, by squadron, when the first aircraft departs home station and when the last aircraft arrives at the beddown base. Also prints only to the screen. Type [shift-Prt Sc] for hard copy.

RECOMMENDED TECHNIQUE . . .

1. Determine which squadrons to deploy and the number of aircraft from each squadron.

2. Determine the route of flight for each squadron. Divide the routings into groups which share one or more legs. From the appropriate meteorological charts (or by calling the weather shop), get the wind factors for all flight legs.

3. Review the aircraft handling capabilities of the bases in the common route segments. The airfield most likely to become a choke point should be designated the control base. Based on the control base's current resources and those that can be airlifted to supplement them, determine the maximum launch frequency the choke point can realistically support and the time the base will be able to support the flow.

Example: C-130 squadrons A, B and C are deploying from Elmendorf, Niagara Falls and Pittsburg, respectively. They will each refuel at Goose Bay then go their separate ways to beddown bases. Squadrons D departs Pope for Lajes, Torrejon and beddown at Aviano. Squadron E departs Dyess for Pope, Lajes and beddown at Torrejon.

Squadrons A, B and C share the refueling stop at Goose Bay and should be grouped together in one flow plan with Goose Bay as the control base. An ALCE team is enroute to

Goose Bay and will be able to launch a mission every thirty minutes, beginning at 1300 on 2 JUN.

Squadrons D and E share the route segments from Pope to Lajes and from Lajes to Torrejon and should be grouped in a second flow plan. Lajes will be designated the control base, with missions departing every forty-five minutes.

4. Initiate the DEPLOY program. When the MAIN MENU appears, select menu item 1, CREATE A NEW FLOW PLAN. The example above deploys only C-130 squadrons. Since the C-130 is one of the aircraft types whose parameters are in the DEPLOY code, there is no reason to select item 1 of the INPUT CHECKLIST. If the plan calls for standard ground times at enroute stops, it will not be necessary to select item 2.

5. Select the FLIGHT ROUTINGS option and enter the routes determined earlier. For this example, routings may be entered for both flow plans (five total). Completion of that option will send you to the WIND FACTORS routine and request the wind factor for each leg.

6. CHECK FLYING TIMES (item eight) to insure they do not exceed the capability of the deploying aircraft. If you find a leg that is questionable, return to FLIGHT ROUTINGS and reroute the mission.

7. Select the MISSION TEMPLATES option and assign each squadron a route of flight and mission number.

8. Select item six and enter CYYR, the ICAO identifier of the choke point for the first flow. The narrow MOG (from the BASE#.DAT file) at CYYR will appear and you can update it if required. You will then be offered a chance to change the wide MOG at CYYR. Enter the time (1300) and date (2 JUN) of CYYR's first departure and the time between departures (45 minutes).

9. RETURN TO MAIN MENU. If you did not select the first two items, you will get a warning flag to the left of item one. Reselect RETURN TO MAIN MENU and you will be allowed to return.

10. Save the information using the SAVE FILE option.

11. Select FLOW MENU and FLOW & MOG the data you have entered. When the mission templates appear, enter the templates for the first flow and the number of aircraft for each squadron.

12. When the FLOW MENU reappears select item four and get a MISSION SCHEDULE from the printer. If it becomes necessary to change ground or flight times, use the CHANGE MISSIONS option. This option will also allow missions to be moved around in the flow.

13. Adjustments may be made with the SLIDE MISSIONS option. If any of the (REFLOW) options are selected from the CHANGE MISSIONS menu, the sliding of missions will be erased and the missions will reflect the control times on the original printed schedule.

14. When you are satisfied with the flow plan, get a final MISSION SCHEDULE and make copies of the FORCE CLOSURE and FUEL REQUIRED (using the [shift-Prt Sc] keys).

I. VARIABLES (range) [hard coded]

AFLDS	number of airfields read from data file
JCB	number corresponding to choke point
JDLA	julian day of first choke point departure
JFI	flow interval in minutes from choke point
MAXACFT	max number of different acft types (1-9)
MAXADDSQ	max number of additions to squad#.dat file [5 per each run of DEPLOY]
MAXAFLD	max number of airfields (50)
MAXBSE	max number of bases in a routing [6]
MAXHOPS	max number of hops in a routing [5]
MAXLEGS	5 * maxmsns (5-2500)
MAXMSNS	max number of missions in one plan (1-500)
MAXPLANS	max number of mission templates [20]
MAXRTES	max number of planned flight routes [20]
NSQUAD	number of squadrons read from data file
NUMACFT	number of acft types used in plan
NUMMSNS	total number of missions in the flow plan

VARIABLES USED THROUGHOUT PROGRAM TO REFER TO IMST ARRAY:

IJET	= 1	type acft
ICTM	= 2	mission control time
IFSTLG	= 3	first leg of mission
ILSTLG	= 7	last . . .
IFSTBSE	= 8	base of origin
ILSTBSE	= 13	destination base
IPRE	= 14	mission time before choke point
ICBGND	= 15	ground time at choke point
IPOST	= 16	mission time after choke point
ILKTO	= 17	which mission this one is linked to (follows)
ILKFRM	= 18	. . . . . from (precedes)
ISQD	= 19	squadron flying this mission
IMSTTOP	= 19	number of columns in IMST array

II. ARRAYS(dimensions)

MISSION ARRAYS

IMST(maxmsns + maxplans, imsttop)

IJET	ICTM	IFSTLG-ILSTLG	IFSTBSE-ILSTBSE	IPRE	ICBGND
1 ACFT TYPE	2 CTRL TIME	3-7 LEG PTRS	8-13 ENRTE BASES	14 PRE-C.Pt. TIME	15 C.Pt. GROUND TIME

IPOST	ILKTO	ILKFRM	ISQD
16 POST-C.Pt. TIME	17 MSN LNKD TO	18 MSN LNKD FROM	19 SQUAD

SMST(maxmsns + maxplans)

MISSION NUMBER

LEGS(maxlegs, 7)

1	2	3	4	5	6	7
IMST BACK PTR	DEPT TIME	DEPT BASE	FLT TIME	DEST GND TIME	DEST BASE	DEST GND FNCT

AIRFIELD ARRAYS

AFLD(aflds, 4)

1	2	3	4
LAT	LONG	NARROW MOG	WIDE MOG

SAFLD(aflds, 2)

1	2
NAME	ICAO

FUEL(aflds)

FUEL REQUIRED FOR PLAN

ROUTES(maxrtes, maxbse)

1	2-5	6
ORIGIN BASE	ENROUTE BASES	DEST BASE

GND TIMES(maxacft, maxbse)

1	2	3	4
ERO TIME	DNLD TIME	UPLD TIME	QUICK TURN

ACFT(maxacft, 3)

1	2	3
BLOCK SPEED	ACFT SIZE	FUEL/ HOUR

SJETS(maxacft)

ACFT ID

WF(maxrtes * maxhops, 3)

1	2	3
DEPT BASE	ARRIVAL BASE	WIND FACTOR

ROUTES(maxrtes, maxbse)

1	2-5	6
ORIGIN BASE	ENROUTE BASES	DEST BASE

DIST(maxrtes, maxhops, 2)

1	2
DEPT BASE	ARRIVAL BASE

[for each route]

LEG DIST (NM)
WIND FACTOR

[for each leg of each route]

FLTTIME(maxrtes, maxbse, numacft)

FLIGHT TIME	for each leg in ROUTES array

[for each type aircraft]

ISQUAD(nsquad + 5, 7)

1	2	3	4	5	6
SQDRN ID	ACT, RES, ANG	TYPE ACFT	PAA	HOME BASE	DAY AVLBL

```
' **********        DEPLOY.BAS          **********
'                   Mike Foster
'                    6 MAR 88
'
CLS : LOCATE 10, 20
PRINT "LOADING SQUADRON AND AIRFIELD DATA"
OPTION BASE 1
DEFINT I-N, P, T: DEFSTR S, Z
'$DYNAMIC
'
MAXMSNS  = 300
MAXPLANS = 20
MAXACFT  = 9
MAXRTES  = 20
MAXADDSQ = 5
'
' variables for IMST array
'
ISUM    = 0 : DISKFLG = 0 : PATCH  = 0
IJET    = 1 : ICTM    = 2 : IFSTLG = 3 : ILSTLG = 7
IFSTBSE = 8 : ILSTBSE = 13: IPRE   = 14
ICBGND  = 15: IPOST   = 16: ILKFRM = 17: ILKTO  = 18
ISQD    = 19: IMSTTOP = 19
'
MAXHOPS = ILSTBSE - IFSTBSE: MAXBSE = MAXHOPS + 1
MAXLEGS = MAXMSNS * MAXHOPS
MAXWF   = MAXRTES * MAXHOPS
'
DIM IMST(MAXMSNS + MAXPLANS, IMSTTOP), SMST(MAXMSNS + MAXPLANS)
DIM LEGS(MAXLEGS, 7), ACFT(MAXACFT, 3)
DIM NDATES(12), NDAYS(12), SNOTE(MAXPLANS)
DIM JARAY(10, 2), NARAY(10, 2)
DIM ROUTES(MAXRTES, MAXBSE), WF(MAXRTES * MAXHOPS, 3)
DIM DIST(MAXRTES, MAXHOPS, 2), SERVICE(3)
DIM SJETS(MAXACFT), GNDTIMES(MAXACFT, 6)
DIM SELEC(10), NSQDPLY(MAXPLANS, 2)
DIM FLTTIME(MAXRTES, MAXHOPS + 1, MAXACFT)
'
ICHGSQ = 0
'
SDOT10 = "..........": SDOT9 = "........."
ZMONTH = "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"
'
DATA 31,28,31,30,31,30,31,31,30,31,30,31
FOR I = 1 TO 12
    READ NDAYS(I)
    NDATES(I) = ISUM
    ISUM = ISUM + NDAYS(I)
NEXT I
'
SERVICE(1)="ACTIVE ":SERVICE(2)="RESERVE":SERVICE(3)="  ANG  "
```

```
'
' read airfield database BASE#.DAT
'
OPEN "I", 2, "BASE#.DAT"
INPUT #2, AFLDS
MXAFLD = AFLDS + 5
DIM AFLD(AFLDS + 5, 4), SAFLD(AFLDS + 5, 2)
DIM FUEL(AFLDS)
FOR I = 1 TO AFLDS
    INPUT #2, SAFLD(I, 1)          ' base
    INPUT #2, SAFLD(I, 2)          ' icao
    INPUT #2, LAT$
       AFLD(I, 1) = VAL(LAT$)      ' latitude
    INPUT #2, LONG$
       AFLD(I, 2) = VAL(LONG$)     ' longitude
    INPUT #2, NMOG$
       AFLD(I, 3) = VAL(NMOG$)     ' narrow mog
    INPUT #2, WMOG$
       AFLD(I, 4) = VAL(WMOG$)     ' wide mog
NEXT I
CLOSE #2
'
'   read squadron database SQUAD#.DAT
'
OPEN "I", 2, "SQUAD#.DAT"
INPUT #2, NSQUAD
DIM ISQUAD(NSQUAD + 5, 7)
DIM FLOWTEMP(NSQUAD)
FOR I = 1 TO NSQUAD
    INPUT #2, SDAT
       ISQUAD(I, 1) = VAL(SDAT) ' squadron
    INPUT #2, SDAT
       ISQUAD(I, 2) = VAL(SDAT) ' 1-usaf,2-afres,3-ang
    INPUT #2, SDAT
       ISQUAD(I, 3) = VAL(SDAT) ' type acft  1-C130ABE 2-C130H
    INPUT #2, SDAT
       ISQUAD(I, 4) = VAL(SDAT) ' paa
    INPUT #2, SDAT
    FOR J = 1 TO AFLDS
       IF SAFLD(J, 2) = SDAT THEN ISQUAD(I,5) = J:EXIT FOR
    NEXT J
    INPUT #2, ISQUAD(I,6)        ' first available day (1,2,3,...)
    INPUT #2, ISQUAD(I, 7)       ' in place nlt day (1,2,3,...)
NEXT I
CLOSE #2
'
```

```
        NUMACFT = 5
        SJETS(1) = 'C130 ': SJETS(2) = 'C130H'
        SJETS(3) = 'C141 ': SJETS(4) = 'C5   ': SJETS(5) = 'KC10 '
        ACFT(1, 1) = 270: ACFT(2, 1) = 285: ACFT(3, 1) = 400
        ACFT(4, 1) = 420: ACFT(5, 1) = 460
        ACFT(1, 3) = 5: ACFT(2, 3) = 5: ACFT(3, 3) = 13
        ACFT(4, 3) = 25: ACFT(5, 3) = 20
        GNDTIMES(1, 1) = 30: GNDTIMES(1, 2) = 90: GNDTIMES(1, 3) = 90
        GNDTIMES(1, 4) = 90: ACFT(1, 2) = 0
        GNDTIMES(2, 1) = 30: GNDTIMES(2, 2) = 90: GNDTIMES(2, 3) = 90
        GNDTIMES(2, 4) = 90: ACFT(2, 2) = 0
        GNDTIMES(3, 1) = 45: GNDTIMES(3, 2) = 135: GNDTIMES(3, 3) = 135
        GNDTIMES(3, 4) = 135: ACFT(3, 2) = 0
        GNDTIMES(4, 1) = 75: GNDTIMES(4, 2) = 195: GNDTIMES(4, 3) = 225
        GNDTIMES(4, 4) = 135: ACFT(4, 2) = 1
        GNDTIMES(5, 1) = 0: GNDTIMES(5, 2) = 180: GNDTIMES(5, 3) = 300
        GNDTIMES(5, 4) = 90: ACFT(5, 2) = 1
        '
        '-----------------------------------------------------------------
        '
500     CLS : PRINT SPC(23); STRING$(37, '_'): PRINT
        PRINT SPC(36); 'MAIN MENU'
        PRINT SPC(23); STRING$(37, '_'): PRINT
        PRINT SPC(23); '1 '; SDOT10; ' CREATE A NEW FLOW PLAN':PRINT
        PRINT SPC(23); '2 '; SDOT10; ' CHANGE CURRENT FLOW PLAN':PRINT
        PRINT SPC(23); '3 '; SDOT10; ' LOAD FILE': PRINT
        PRINT SPC(23); '4 '; SDOT10; ' SAVE FILE': PRINT
        PRINT SPC(23); '5 '; SDOT10; ' CHANGE SQUADRON DATA': PRINT
        PRINT SPC(23); '6 '; SDOT10; 'ADD AIRFIELD TO DATA BASE':PRINT
        PRINT SPC(23); '7 '; SDOT10; SDOT10; ' FLOW MENU': PRINT
        PRINT SPC(23); '8 '; SDOT10; ' END'
        LOCATE 23, 32: INPUT 'ENTER REQUEST  '; IANSW
        CLS
        IF IANSW < 1 OR IANSW > 8 THEN 500
700     IF IANSW = 8 THEN 61000
        ON IANSW GOSUB 710, 760, 4360, 5650, 60000, 62000, 20510, 700
        GOTO 500
        '
```

```
'-----------------------------------------------
'  called from main menu
'
710   LOCATE 10, 30: PRINT "CLEARING ARRAYS . . ."
      FOR II = 1 TO 7: SELEC(II) = "": NEXT II
      LSTCHNC = 0: NUMACFT = 5: ICHBSE = 0
      FOR I = 1 TO MAXMSNS + MAXPLANS
          SMST(I) = ""
          FOR II = 1 TO IMSTTOP
             IMST(I, II) = 0
          NEXT II
      NEXT I
      '
      FOR I = 1 TO MAXLEGS
          FOR II = 1 TO 7
             LEGS(I, II) = 0
          NEXT II
      NEXT I
      '
      FOR I = 6 TO MAXACFT
          SJETS(I) = ""
          FOR II = 1 TO 3
             ACFT(I, II) = 0
          NEXT II
          FOR III = 1 TO 6
             GNDTIMES(I, III) = 0
          NEXT III
      NEXT I
      '
      FOR I = 1 TO MAXRTES
          FOR J = 1 TO MAXBSE
             ROUTES(I, J) = 0
          NEXT J
          FOR II = 1 TO MAXHOPS
             FOR J = 1 TO 2
                DIST(I, II, J) = 0
             NEXT J
             FOR J = 1 TO MAXACFT
                FLTTIME(I, II, J) = 0
             NEXT J
          NEXT II
      NEXT I
      '
      FOR I = 1 TO MAXRTES * MAXHOPS
          FOR II = 1 TO 3
             WF(I, II) = 0
          NEXT II
      NEXT I
      '
      FOR I = 1 TO MAXPLANS
          SNOTE(I) = ""
      NEXT I
```

```
      '
      GOTO 800
      '
760   FOR II = 1 TO 6: SELEC(II) = "**": NEXT II
      '
800   CLS : LOCATE 1, 25: PRINT "INPUT CHECKLIST ( COMPLETED (**) )"
      PRINT : PRINT
      PRINT SPC(30); "1 ..... AIRCRAFT TYPES": PRINT
      PRINT SPC(30); "2 ..... GROUND TIMES": PRINT
      PRINT SPC(30); "3 ..... FLIGHT ROUTINGS": PRINT
      PRINT SPC(30); "4 ..... ROUTE WIND FACTORS": PRINT
      PRINT SPC(30); "5 ..... MISSION TEMPLATES": PRINT
      PRINT SPC(30); "6 ..... CHOKE POINT, TIME, DATE": PRINT
      PRINT SPC(30); "7 ..... REVIEW AIRFIELDS": PRINT
      PRINT SPC(30); "8 ..... CHECK FLYING TIMES": PRINT
      PRINT SPC(30); "9 ..... RETURN TO MAIN MENU"
      '
      Y = 4
      FOR I = 1 TO 6
          LOCATE Y, 25: PRINT "("; SELEC(I); ")"
          Y = Y + 2
      NEXT I
900   LOCATE 23, 35: INPUT "ENTER REQUEST  "; IANSW
      IF IANSW < 1 OR IANSW > 9 THEN 800
      IF IANSW < 9 THEN LSTCHNC = 0: GOTO 1000
      IF LOADFLG = 1 THEN 1060
      DUMBFLG = 0: DISKFLG = 0
      '
      FOR I = 1 TO 6
          IF SELEC(I) <> "**" THEN DUMBFLG = I: EXIT FOR
      NEXT I
      '
      IF DUMBFLG = 0 OR LSTCHNC = 1 THEN 1060
      Y = (DUMBFLG * 2): LOCATE Y + 2, 1
      PRINT "INCOMPLETE ---------->": LSTCHNC = 1: GOTO 900
1000  SELEC(IANSW) = "**"
1040  ON IANSW GOSUB 1530, 1860, 2390, 5301, 3020, 1110, 4970, 5500
      GOTO 800
1060  RETURN      ' TO 650
```

```
'
' JCB  choke point
'      called from input menu
'
1110  CLS : JCB = 0
      PRINT SPC(15); "DEPARTURES ARE REGULATED FROM CHOKE POINT"
      PRINT : PRINT
      PRINT SPC(8);"ENTER ICAO OF CHOKE POINT              ";
      INPUT SANSW
      FOR I = 1 TO AFLDS
         IF SAFLD(I, 2) = SANSW THEN JCB = I
      NEXT I
      IF JCB < 1 OR JCB > AFLDS THEN BEEP: GOTO 1110
      '
      PRINT : PRINT SPC(8); "NARROW MOG "; AFLD(JCB, 3)
      LOCATE 6, 28
      INPUT "'ENTER' IF OK, 'C' TO CHANGE    "; SANSW1
      IF SANSW1 = "C" OR SANSW1 = "c" THEN
         ICHBSE = 1
        PRINT SPC(28);"ENTER NARROW MOG AT";SAFLD(JCB,2);"     ";
         INPUT AFLD(JCB, 3)
      END IF
      '
      PRINT : PRINT SPC(8); "WIDE MOG    "; AFLD(JCB, 4)
      LOCATE 8, 28
      INPUT "'ENTER' IF OK, 'C' TO CHANGE    "; SANSW1
      IF SANSW1 = "C" OR SANSW1 = "c" THEN
          ICHBSE = 1
          PRINT SPC(28); "ENTER WIDE MOG AT "; SAFLD(JCB, 2);"     ";
          INPUT AFLD(JCB, 4)
      END IF
```

```
      '
      ' JLAT  first takeoff time from choke point
1200  '
1210  PRINT : PRINT SPC(8);
      PRINT "ENTER FIRST T.O. TIME FROM ";
      PRINT SAFLD(JCB, 2); " (HHMM)              ";
      INPUT SANSW
      IF LEN(SANSW) <> 4 THEN BEEP: GOTO 1210
      IF VAL(LEFT$(SANSW, 2)) > 24 OR VAL(RIGHT$(SANSW, 2)) > 59 THEN
         BEEP: GOTO 1210
      END IF
      SHR = LEFT$(SANSW, 2): IHR = VAL(SHR) * 60
      SMN = RIGHT$(SANSW, 2): IMN = VAL(SMN)
      ITP = IHR + IMN
      JLAT = ITP
1300  PRINT : PRINT SPC(8);
      INPUT "ENTER DATE (DDMMYY) OF T.O.                          "; SDATE
      IF LEN(SDATE) <> 6 THEN BEEP: GOTO 1300
      IF VAL(LEFT$(SDATE, 2)) > 31 OR VAL(MID$(SDATE, 3, 2)) > 12 THEN
         BEEP: GOTO 1300
      END IF
      SDTE = SDATE
      '
      ' calendar -> julian
      '
      IDY = VAL(LEFT$(SDATE, 2))
      IMO = VAL(MID$(SDATE, 3, 2)): IYR = VAL(RIGHT$(SDTE, 2))
      IF IYR MOD 4 = 0 THEN LEAP = 1 ELSE LEAP = 0
      IF IYR MOD 4 = 1 THEN LSTLEAP = 1 ELSE LSTLEAP = 0
      JDLA = NDATES(IMO) + IDY: IF JDLA > 60 THEN JDLA = JDLA + LEAP
      '
      '  JFI  choke point takeoff interval
      '
      PRINT : PRINT SPC(8);
      PRINT "ENTER INTERVAL BETWEEN "; SAFLD(JCB, 2);
      PRINT " TAKEOFFS (IN MINUTES)  ";
      INPUT JFI
      '
      GOSUB 4270: IF SANSW <> "Y" AND SANSW <> "y" THEN 1110
      RETURN
```

```
'
'  number and type of acft
'  called from input menu
'
1530  CLS : CHNGFLG = 0: LOCATE 6, 10
      PRINT SPC(9); "PROGRAM CONTAINS PARAMETERS FOR THESE AIRCRAFT"
      PRINT : PRINT SPC(18); STRING$(45, "_")
      LOCATE 10, 19
      PRINT "C130A/B/E     C130H     C141     C5     KC10"
      PRINT SPC(18); STRING$(45, "_")
      IF NUMACFT > 5 THEN
         PRINT : PRINT SPC(19);
         FOR I = 6 TO NUMACFT
            PRINT SJETS(I),
         NEXT I
         PRINT : PRINT SPC(18); STRING$(45, "_")
      END IF
      PRINT : PRINT
      IF NUMACFT = 9 THEN
         PRINT SPC(30);
         INPUT "'ENTER' TO CONTINUE"; SS: GOTO 1840
      ELSE
         PRINT SPC(18);
         INPUT "'ENTER' TO CONTINUE, 'A' TO ADD AIRCRAFT  "; SANSW
      END IF
      IF SANSW <> "A" AND SANSW <> "a" THEN 1840
1680  CLS : CHNGFLG = 1: SELEC(4) = ""
      LOCATE 2, 10
      PRINT"ENTER # OF AIRCRAFT TYPES TO BE ADDED (";MAXACFT - NUMACFT;
      INPUT " MAX )"; IANSW
      IF IANSW = 0 THEN RETURN
      IF IANSW > 5 THEN BEEP: GOTO 1680
      CLS : PRINT "     ACFT ID     WIDE BODY (Y/N)";
      PRINT "      BLOCK SPEED     FUEL/HOUR (1000 #)"
      PRINT
```

```
      FOR I = 1 TO IANSW
          PRINT
          PRINT I; '      ': Y = CSRLIN - 1: X = POS(I)
1750      LOCATE Y, X+5: INPUT SANSW:IF SANSW = '' THEN BEEP: GOTO 1750
          SJETS(NUMACFT + I) = LEFT$(SANSW + '      ', 5)
1770      '
          LOCATE Y, X + 25
          INPUT SANSW
          ITEMP = 0
          IF SANSW <> 'Y' AND SANSW <> 'N' THEN ITEMP = 1
          IF ITEMP=1 AND SANSW<>'y' AND SANSW<>'n' THEN BEEP: GOTO 1770
          IF SANSW = 'Y' THEN
            ACFT(NUMACFT + I, 2) = 1
          ELSE
            ACFT(NUMACFT + I, 2) = 0
          END IF
1775      '
          LOCATE Y, X + 37
          INPUT ITEMP
          IF ITEMP = 0 THEN LOCATE Y,X+37: PRINT '         ':GOTO 1775
          ACFT(NUMACFT + I, 1) = ITEMP ' block speed
1780      '
          LOCATE Y, X + 55
          INPUT ITEMP
          IF ITEMP = 0 THEN LOCATE Y, X+55:PRINT'          ': GOTO 1780
          ACFT(NUMACFT + I, 3) = ITEMP  'fuel flow
      NEXT I
      GOSUB 4270: IF SANSW <> 'Y' AND SANSW <> 'y' THEN 1680
      NUMACFT = NUMACFT + IANSW
      IF CHNGFLG = 1 THEN GOSUB 1860
1840  RETURN
```

```
'
'    ground times
'    called from input menu
'
1860  CLS : LOCATE 1, 23
      PRINT " ERO      DNLOAD    UPLOAD   EN ROUTE"
      PRINT SPC(23); STRING$(36, "-")
      Y = 3: X = 13
      FOR I = 1 TO NUMACFT
          LOCATE Y, (X - 5)
          PRINT I; "     "; LEFT$(SJETS(I) + "    ", 6); "    ";
          FOR J = 1 TO 4
             SHR = RIGHT$(STR$(INT(GNDTIMES(I, J) / 60) + 100),2)+":"
             SMIN = RIGHT$(STR$((GNDTIMES(I, J) MOD 60) + 100),2)
             SGND = SHR + SMIN
             PRINT SGND; "      ";
          NEXT J
          Y = Y + 2
      NEXT I
      LOCATE Y - 1, 9: PRINT STRING$(50, "_")
      LOCATE Y + 1, 15
      PRINT "'C' TO CHANGE ONE OR MORE GROUND TIMES,"
      LOCATE Y + 2, 19: INPUT "'ENTER' TO RETURN TO MENU  "; SANSW
      IF SANSW <> "C" AND SANSW <> "c" THEN 2340
2060  LOCATE Y + 2, 5: PRINT SPC(70); ""
      LOCATE Y + 1, 10
      INPUT "           ENTER NUMBER OF THE ONE TO CHANGE"; IANSW
      IF IANSW < 1 OR IANSW > NUMACFT THEN
         BEEP: LOCATE (Y + 1), 1
         PRINT SPC(60); : GOTO 2060
      END IF
      LOCATE (Y + 1), 10: PRINT SPC(69); : LOCATE (Y + 1), 20
      PRINT "ENTER ALL TIMES USING 4 DIGITS (HHMM)"
      PRINT SPC(20); "OR HIT 'ENTER' TO USE EXISTING TIME"
      Y = (2 * IANSW + 1): X = 23
```

```
      FOR II = 1 TO 4
2160     LOCATE Y, X
          INPUT SANSW
          IF SANSW <> '' THEN 2240
          IHR = INT(GNDTIMES(IANSW, II) / 60) * 60
          IMIN = GNDTIMES(IANSW, II) MOD 60
          SHR = RIGHT$(STR$(INT(GNDTIMES(IANSW, II) / 60) + 100), 2)
          SMIN = RIGHT$(STR$((GNDTIMES(IANSW, II) MOD 60) + 100), 2)
          SANSW = SHR + SMIN: GOTO 2290
2240      IF LEN(SANSW) <>4 THEN
           LOCATE Y,X:PRINT '     ': BEEP: GOTO 2160
          END IF
          IF VAL(LEFT$(SANSW, 2)) > 24 THEN
            LOCATE Y, X: PRINT '     ': BEEP: GOTO 2160
          END IF
          IF VAL(RIGHT$(SANSW, 2)) > 59 THEN
            LOCATE Y, X: PRINT '     ': BEEP: GOTO 2160
          END IF
          IMIN = VAL(RIGHT$(SANSW, 2))
          IHR = VAL(LEFT$(SANSW, 2)) * 60
2290      GNDTIMES(IANSW, II) = IMIN + IHR
          LOCATE Y,X+1
          PRINT LEFT$(SANSW, 2)+ '+' +RIGHT$(SANSW, 2)
          X = X + 10
      NEXT II
      GOTO 1860
2340  SELEC(2) = '**': RETURN
```

```
'
'   flight routes
'   called from input menu
'
2390  CLS : CHNGFLG = 0: PRINT '       ';
      PRINT 'ENTER ICAOs OF PLANNED ROUTE OF FLIGHT'
      Y = 2: X = 10
      FOR I = 1 TO MAXRTES
          LOCATE Y, 5
          PRINT I;
          IF I < 10 THEN PRINT ' ';
          FOR J = 1 TO MAXBSE
             LOCATE Y, X
            IF ROUTES(I,J)<>0 THEN PRINT SAFLD(ROUTES(I,J), 2) ;'       '
             X = X + 10
          NEXT J
          Y = Y + 1: X = 10
      NEXT I
2520  LOCATE 22, 8
      INPUT 'ENTER THE NUMBER OF ROUTING TO CHANGE (ENTER TO END)';SANSW
      IF SANSW <> '' THEN CHNGFLG = 1
      IANSW = VAL(SANSW): LOCATE 22, 68: PRINT '      '
      IF SANSW = '' THEN 2920
      IF IANSW < 1 OR IANSW > MAXRTES THEN BEEP: GOTO 2520
      LOCATE 22, 8: PRINT SPC(71);
      GOSUB 2730     ' clear old route
      Y = 1 + IANSW: X = 10: LOCATE Y, X - 2: PRINT SPC(60);
'
'   enter icaos
'
      FOR J = 1 TO MAXBSE
2590     LOCATE Y, X - 2
          IOK =0: INPUT ZANSW: IF ZANSW = '' THEN GOSUB 2830: GOTO 2670
          FOR K = 1 TO AFLDS
             IF ZANSW = SAFLD(K, 2) THEN
               ROUTES(Y - 1, INT(X / 10)) = K: IOK = 1: EXIT FOR
             END IF
          NEXT K
          IF IOK = 1 THEN LOCATE Y, X - 2: PRINT '   '; ZANSW
          IF IOK = 0 OR LEN(ZANSW) <> 4 THEN
            BEEP: LOCATE Y, X: PRINT '          ': GOTO 2590
          END IF
2670     X = X + 10
      NEXT J
      GOSUB 2962 'enter leg into WF array
      GOTO 2520
```

```
      '
      ' called from 2560
      '
2730  FOR JJ = 1 TO MAXBSE
          ROUTES(IANSW, JJ) = 0
      NEXT JJ
      RETURN     ' TO 2560
      '
      ' called from 2600
      '
2830  FOR JJ = J TO MAXBSE
          ROUTES(IANSW, JJ) = 0
      NEXT JJ
      J = MAXBSE
      RETURN     ' TO 2600
      '
2920  JSEQ = 0
      FOR J1 = 1 TO 10
          IF ROUTES(J1, 1) <> 0 THEN JSEQ = JSEQ + 1 ELSE J1 = 10
      NEXT J1
      SELEC(4) = "**": GOSUB 2980: IF CHNGFLG = 1 THEN GOSUB 5301
      RETURN     ' TO 1040
      '
2962  '  enter leg into WF array
      '
      FOR K = 1 TO MAXHOPS
          IF ROUTES(IANSW, K) = 0 OR ROUTES(IANSW, K + 1) = 0 THEN 2972
          FOR L = 1 TO MAXWF
             IF ROUTES(IANSW, K) = WF(L, 1) THEN
               IF ROUTES(IANSW, K + 1) = WF(L, 2) THEN EXIT FOR
             END IF
             IF WF(L, 1) = 0 THEN
               WF(L, 1) = ROUTES(IANSW, K)
               WF(L, 2) = ROUTES(IANSW, K + 1)
               EXIT FOR
             END IF
2970      NEXT L
2972  NEXT K
      RETURN     ' TO 2685
```

```
'
' dist array
'
2980  CLS
      FOR I = 1 TO MAXRTES
          FOR J = 1 TO MAXHOPS
             DIST(I, J, 1) = 0    'DISTANCE
             DIST(I, J, 2) = 0    'WIND FACTOR
          NEXT J
      NEXT I
      FOR I = 1 TO MAXRTES
          FOR J = 1 TO MAXHOPS
             IF ROUTES(I, J + 1) = 0 THEN EXIT FOR
            RLAT1 = AFLD(ROUTES(I, J), 1)
            RLONG1 = AFLD(ROUTES(I, J), 2)
             RLAT2 = AFLD(ROUTES(I, J + 1), 1)
             RLONG2 = AFLD(ROUTES(I, J + 1), 2)
             GOSUB 30920  ' calculate great circle distance
             DIST(I, J, 1) = DIST
          NEXT J
      NEXT I
      RETURN
'
'  mission templates IMST(TOP 20)
'  called from input menu
'
3020  CLS : Y = 3: X = 5
      LOCATE 1, 30: PRINT "MISSION PLANS"
      FOR I = (MAXMSNS + 1) TO (MAXMSNS + MAXPLANS)
          LOCATE Y, X
          PRINT I - MAXMSNS
          IF IMST(I, IJET) = 0 THEN
            NUMTEMPS = I - MAXMSNS - 1
            Y = Y + 2
            LOCATE Y, 3
            EXIT FOR
          END IF
          LOCATE Y, (X + 5): PRINT "  "; SJETS(IMST(I, IJET));"  ";
          FOR J = IFSTBSE TO ILSTBSE
             IF IMST(I, J) = 0 THEN EXIT FOR
             PRINT SAFLD(INT(IMST(I, J) / 100), 2); "  ";
          NEXT J
          LOCATE Y, 55
          PRINT ISQUAD(IMST(I, ISQD), 1); " TAS  "; SNOTE(I - MAXMSNS)
          PRINT
3140      Y = Y + 1
      NEXT I
3160  INPUT "        ENTER TEMPLATE TO CHANGE, HIT ENTER TO END ";IANSW1
      IF IANSW1 = 0 THEN RETURN
      IF IANSW1 < 1 OR IANSW1 > MAXPLANS THEN BEEP: GOTO 3160
'
      CLS : LOCATE 10, 20: INPUT "ENTER SQUADRON  "; ISQ
```

```
'
'   see if squadron is in data base
'
PTR = 0    ' identifies squadron
FOR I = 1 TO NSQUAD
    IF ISQUAD(I, 1) = ISQ THEN PTR = I: EXIT FOR
NEXT I
IF PTR = 0 THEN
   CLS : LOCATE 8, 1
   PRINT 'THE SQUADRON YOU ENTERED IS NOT IN THE DATA BASE'
   PRINT
   PRINT ' TO USE THAT SQUADRON YOU MUST RETURN TO MAIN MENU'
   PRINT
   PRINT ' AND CHANGE THE DATA BASE ( MENU ITEM 5 )'
   PRINT
   INPUT ' HIT 'ENTER' TO CONTINUE'; TEMP
   GOTO 3020
END IF
'
'   see if route exists for that squadron
'
IOK = 0
FOR I = 1 TO MAXRTES
    IF ROUTES(I, 1) = 0 THEN EXIT FOR
    IF ROUTES(I, 1) = ISQUAD(PTR, 5) THEN IOK = 1: EXIT FOR
NEXT I
IF IOK = 0 THEN
   CLS : LOCATE 8, 1
   PRINT 'THERE ARE NO ROUTES FOR THE SQUADRON ENTERED'
   PRINT
   PRINT ' TO USE THAT SQUADRON YOU MUST RETURN TO INPUT MENU'
   PRINT
   PRINT ' AND ENTER A ROUTE ( MENU ITEM 4 )'
   PRINT
   INPUT ' HIT 'ENTER' TO CONTINUE'; TEMP
   GOTO 3020
END IF
'
'   erase old mission template
'
FOR I = 1 TO IMSTTOP
    IMST(MAXMSNS + IANSW1, I) = 0
NEXT I
SMST(MAXMSNS + IANSW1) = ''
'
IMST(MAXMSNS + IANSW1, ISQD) = PTR
```

```
'
' print routings
'
3170  CLS : Y = 2: X = 5
      JSEQ = 0   ' number of routes
      CTR = 0    '
      CTR2 = 0   '
      FOR J = 1 TO MAXRTES
          LOCATE Y, X: PRINT J
          FOR K = 1 TO MAXBSE
             IF ROUTES(J, 1) = 0 THEN                    ' route empty
               LOCATE Y, X: PRINT '     ': J = MAXRTES: EXIT FOR
             END IF
             IF ROUTES(J, 1) <> ISQUAD(PTR, 5) THEN   ' not sq base
               LOCATE Y, X: PRINT '     ': Y = Y - 1: EXIT FOR
             END IF
             IF ROUTES(J, K) = 0 OR K = MAXBSE THEN
               CTR2 = CTR2 + 1: CTR3 = J: EXIT FOR
             END IF
             LOCATE Y, X + 5: PRINT SAFLD(ROUTES(J, K), 2)
             CTR = CTR + 1
             X = X + 7
3340      NEXT K
           Y = Y + 1: X = 5: JSEQ = JSEQ + 1: PRINT
      NEXT J
      IF CTR2 = 1 THEN IANSW2 = CTR3: GOTO 3430
3370  LOCATE 23, 10: INPUT 'ENTER DESIRED ROUTING  '; IANSW2
      IF IANSW2 = 0 THEN 3370
      IF ROUTES(IANSW2, 1) <> 0 THEN 3400
      LOCATE 23,10:PRINT '                                  ':BEEP: GOTO 3370
3400  IF IANSW2 < 0 OR IANSW2 > JSEQ THEN BEEP: GOTO 3370
      IF IANSW2 = 0 THEN 3020
      CLS
3430  IMST(IANSW1 + MAXMSNS, IJET) = ISQUAD(PTR, 3)
      SMST(IANSW1 + MAXMSNS) = '00000'
      PRINT
3490  LOCATE 10, 20
      INPUT 'ENTER MISSION # (7 DIGITS MAX)  '; SANSW'
      IF LEN(SANSW)>7 THEN LOCATE 11,10:PRINT SPC(45);:BEEP: GOTO 3490
      SANSW = SANSW + '000000000000'
      SMST(IANSW1 + MAXMSNS) = LEFT$(SANSW, 12)
3530  I = IANSW1 + MAXMSNS
      ANSW = IANSW2
      '
```

```
3560  LOCATE 12, 20
      INPUT "WILL ALL ENROUTE STOPS QUICKTURN (Y/N)  "; SANSW
      IF SANSW = "" THEN BEEP: GOTO 3560
      IF SANSW = "N" OR SANSW = "n" THEN 3670
      QFLAG = 1: SNOTE(IANSW1) = "* QKTRN *"
      FOR J = 1 TO MAXBSE
          IF J = MAXBSE THEN 3640
          IF ROUTES(ANSW, J + 1) > 0 THEN
            IMST(I, J + 7) = (100 * ROUTES(ANSW, J)) + 4: GOTO 3650
          END IF
3640      IMST(I, J + 7) = (100 * ROUTES(ANSW, J)) + 6: J = MAXBSE
3650  NEXT J
      GOTO 3020  '-------NEXT MISSION
3670  CLS
      PRINT"     ERO = 1   DOWNLOAD = 2   UPLOAD = 3   QUICK TURN = 4"
      Y = 8: X = 29
      LOCATE 5, 29: PRINT "GND "
      LOCATE 6, 29: PRINT "FCT "
      NSEQ = 0: Y = 8: X = 41
      FOR J = 1 TO MAXBSE'--------PRINT ICAO
          IF ROUTES(ANSW, J) = 0 THEN J = MAXBSE: GOTO 3780
          IMST(I, 7 + J) = (100 * ROUTES(ANSW, J))
          LOCATE Y, X: PRINT SAFLD(ROUTES(ANSW, J), 2)
          Y = Y + 2
          IF Y < 23 THEN 3780 ELSE PRINT : PRINT : Y = Y - 2
3780  NEXT J
      K = ANSW: Y = 8: X1 = 29: X2 = 35: X3 = 48
      FOR J = 1 TO MAXBSE
          IFCT = 0
          IF J = 1 THEN IFCT = 5: GOTO 3890
          IF J = MAXBSE THEN IFCT = 6: GOTO 3890
          IF ROUTES(K, J + 1) = 0 THEN IFCT = 6: GOTO 3890
3850      LOCATE Y, X1,1: INPUT SFCT: IF SFCT = "" THEN BEEP: GOTO 3850
          IF LEN(SFCT) > 1 THEN
            LOCATE Y, X1: PRINT "            ": BEEP: GOTO 3850
          END IF
          IFCT = VAL(SFCT)
          IF (IFCT < 1) OR (IFCT > 5) THEN
            LOCATE Y, X1, 1: PRINT "   ": BEEP: GOTO 3850
          END IF
3890      IMST(I, J + 7) = IMST(I, J + 7) + IFCT
          IF IFCT = 6 THEN J = MAXBSE: GOTO 3920
          Y = Y + 2
3920  NEXT J
      '
      PRINT
      GOSUB 4270: IF SANSW <> "Y" AND SANSW <> "y" THEN 3670
      GOTO 3020
```

```
'
' called from 1470,1830,3950
'
4270  PRINT : PRINT : PRINT SPC(16);
      INPUT "IS THE ABOVE INFORMATION CORRECT (Y/N)"; SANSW
      IF SANSW = "Y" OR SANSW = "N" THEN RETURN
      IF SANSW = "y" OR SANSW = "n" THEN RETURN ELSE BEEP: GOTO 4270
      '
      ' read file from disk
      ' called from main menu
      '
4360  CLS : PRINT STRING$(80, "_")
      PRINT : FILES
      PRINT STRING$(80, "_"): PRINT : PRINT
      PRINT "DON'T FORGET THE DISK DRIVE SPECIFIER, IF NEEDED"
      PRINT : PRINT "ENTER FILE NAME (WITHOUT SUFFIX) . . .": PRINT
      INPUT " OR 'ENTER' TO RETURN TO MENU"; DFILE$
      IF DFILE$ = "" THEN RETURN
      CLS : PRINT SPC(14); "LOADING "; DFILE$; "" FROM DISK..."
      OPEN "I", 2, DFILE$ + ".DAT"
      INPUT #2, NUMACFT
      FOR I = 1 TO AFLDS
         INPUT #2, AFLD(I, 1): INPUT #2, AFLD(I, 2)
         INPUT #2, AFLD(I, 3)
         INPUT #2, AFLD(I, 4): INPUT #2, SAFLD(I, 1)
         INPUT #2, SAFLD(I, 2)
      NEXT I
      FOR I = 1 TO MAXWF
          INPUT #2, WF(I, 1): INPUT #2, WF(I, 2): INPUT #2, WF(I, 3)
      NEXT I
      FOR I = 1 TO MAXRTES
          FOR J = 1 TO MAXBSE
             INPUT #2, ROUTES(I, J)
          NEXT J
      NEXT I
      INPUT #2, NUMMSNS, NUMTEMPS
      FOR I = 1 TO MAXMSNS + MAXPLANS
          FOR J = 1 TO IMSTTOP
             INPUT #2, IMST(I, J)
          NEXT J
      NEXT I
      INPUT #2, PKEY
      FOR I = 1 TO MAXLEGS
          FOR J = 1 TO 7
             INPUT #2, LEGS(I, J)
         NEXT J
      NEXT I
      INPUT #2, SDTE
```

```
FOR I = 1 TO NUMACFT
    INPUT #2, ACFT(I, 1), ACFT(I, 2), ACFT(I, 3)
    INPUT #2, SJETS(I)
    FOR J = 1 TO 6: INPUT #2, GNDTIMES(I, J): NEXT J
NEXT I
INPUT #2, JLAT, JDLA, JTC, JCB, JFI, JDOC, LEAP, LSTLEAP
FOR I = 1 TO MAXPLANS
    INPUT #2, SNOTE(I)
NEXT I
FOR I = 1 TO MAXMSNS + MAXPLANS
    INPUT #2, SMST(I)
NEXT I
FOR I = 1 TO MAXRTES
    FOR J = 1 TO MAXHOPS
       INPUT #2, DIST(I, J, 1)
       INPUT #2, DIST(I, J, 2)
    NEXT J
NEXT I
FOR I = 1 TO MAXRTES
    FOR J = 1 TO MAXHOPS
       FOR K = 1 TO NUMACFT
          INPUT #2, FLTTIME(I, J, K)
       NEXT K
    NEXT J
NEXT I
INPUT #2, NSQDEP, ALPHA, OMEGA
FOR I = 1 TO MAXPLANS
    INPUT #2, NSQDPLY(I, 1), NSQDPLY(I, 2)
NEXT I
CLOSE #2
RETURN
```

```
      '
      '   icaos and base names
      '   called from input menu
      '
4970  CLS
      II = AFLDS MOD 20: JJ = AFLDS \ 20
      IF II > 0 THEN II = 1
      JJ = JJ + II
      CTR = 0
      FOR I = 1 TO JJ
          CLS : PRINT "ICAO          NAME",
          PRINT SPC(11); "NARROW MOG     WIDE MOG"
          PRINT "____          ----------"; SPC(15);
          PRINT "----------     --------"
          FOR J = 1 TO 20
             IF (J + CTR) > AFLDS THEN EXIT FOR
             PRINT SAFLD(J + CTR, 2),
             PRINT LEFT$(SAFLD(J + CTR, 1) + "          ", 16),
             PRINT AFLD(J + CTR, 3), AFLD(CTR + J, 4)
          NEXT J
          CTR = CTR + 20
4990      LOCATE 23, 20
          INPUT "'C' TO CONTINUE, 'ENTER' TO EXIT  "; SANSW
          IF SANSW <> "" AND SANSW <> "C" AND SANSW <> "c" THEN 4990
          IF SANSW = "" THEN EXIT FOR
      NEXT I
      RETURN
```

```
'
' ------- wind factors WF(100,3)
'         called from input menu
'
5301  CLS : PRINT SPC(30); "WIND FACTORS"
      SHDR = "FROM       TO       WF"
      Y = 3: X = 20
      LOCATE 1, X + 5: PRINT SHDR: PRINT
      FOR I = 1 TO MAXWF
          IF Y MOD 20 = 1 THEN
            Y = 3: X = 20: CLS : LOCATE 1, X + 5: PRINT SHDR
          END IF
          IF WF(I, 1) = 0 THEN 5420
          ' see if entered in opposite direction
          FOR J = 1 TO I - 1
             IF WF(J, 2) = WF(I, 1) AND WF(J, 1) = WF(I, 2) THEN
               WF(I, 3) = -WF(J, 3)
             END IF
          NEXT J
          '
          LOCATE Y, X
          IF I < 10 THEN PRINT " ";
          PRINT I; " "; SAFLD(WF(I, 1), 2); "  TO  ";
          PRINT SAFLD(WF(I, 2), 2); "  =  "; WF(I, 3)
          LOCATE 22, 10
          PRINT"ENTER NEW WIND FACTOR FOR THIS LEG,'ENTER' IF NO CHANGE"
          LOCATE Y, X + 30: INPUT SWF
          IF SWF = "" THEN 5363
          LOCATE Y, X + 27: PRINT SPC(8);
          WF(I, 3) = VAL(SWF)
5363     LOCATE Y, X
          IF I < 10 THEN PRINT " ";
          PRINT I; " "; SAFLD(WF(I, 1), 2); "  TO  ";
          PRINT SAFLD(WF(I, 2), 2); "  =  "; WF(I, 3)
          Y = Y + 1
5420  NEXT I
5430  CLS : LOCATE 10, 10
      INPUT "'R' TO REVIEW WIND FACTORS, 'ENTER' TO EXIT "; SANSW
      IF SANSW = "R" OR SANSW = "r" THEN 5301
      IF SANSW <> "" THEN 5430
```

```
'
'   put wind factor into DIST(I,J,2)
'
CLS : LOCATE 10, 20
PRINT "CALCULATING GROUNDSPEEDS . . ."
FOR I = 1 TO MAXRTES
    FOR J = 1 TO MAXHOPS
       IF ROUTES(I, J) = 0 THEN EXIT FOR
       FOR K = 1 TO MAXWF
          IF ROUTES(I, J) = WF(K, 1) THEN
             IF ROUTES(I, J + 1) = WF(K, 2) THEN
               DIST(I, J, 2) = WF(K, 3)
             END IF
          END IF
       NEXT K
    NEXT J
NEXT I
'
' build flight time array
'
FOR I = 1 TO NUMACFT
    FOR J = 1 TO MAXRTES
       FOR K = 1 TO MAXHOPS
          IF DIST(J, K, 1) = 0 THEN EXIT FOR
         ' blockspd + wind factor
          GNDSPD = ACFT(I, 1) + DIST(J, K, 2)
          FTEMP = INT(((DIST(J, K, 1) / GNDSPD) * 60) + 4)
          '
         ' round up to next 5-min
          '
          MTEMP = INT(((DIST(J, K, 1) / GNDSPD) * 60) + 4) MOD 5
          FLTTIME(J, K, I) = FTEMP - MTEMP
       NEXT K
    NEXT J
NEXT I
GOSUB 5500
RETURN
```

```
'
5500  ' review flying times
'
      FOR I = 1 TO NUMACFT
          CLS : PRINT "-> "; SJETS(I); " <-": PRINT
          FOR J = 1 TO MAXRTES
             FOR K = 1 TO MAXHOPS
                IF DIST(J, K, 1) = 0 THEN EXIT FOR
                ITMP2 = FLTTIME(J, K, I): GOSUB 29770
                PRINT SAFLD(ROUTES(J, K), 2); " "; STME2; "   ";
                IF FLTTIME(J, K + 1, I) = 0 THEN
                  PRINT SAFLD(ROUTES(J, K + 1), 2);
                   EXIT FOR
                END IF
                IF K = MAXHOPS THEN
                   PRINT SAFLD(ROUTES(J, K + 1), 2); : EXIT FOR
                END IF
             NEXT K
             PRINT
          NEXT J
          INPUT "'ENTER' TO CONTINUE"; ANSW
      NEXT I
      RETURN
'
'  save file on disk
'  called from main menu
'
5650  PRINT : PRINT STRING$(80, "_")
      PRINT : FILES: PRINT STRING$(80, "_")
      PRINT : PRINT"DON'T FORGET THE DISK DRIVE SPECIFIER, IF NEEDED"
      PRINT : PRINT "ENTER FILE NAME ( WITHOUT PREFIX ) . . ."
      PRINT : INPUT "OR 'ENTER' TO RETURN TO MENU"; DFILE$
      IF DFILE$ = "" THEN RETURN
      CLS : PRINT : PRINT : PRINT "SAVING "; DFILE$; "..."
      OPEN "O", 2, DFILE$ + ".DAT"
      PRINT #2, NUMACFT
      FOR I = 1 TO AFLDS
          PRINT #2, AFLD(I,1): PRINT #2, AFLD(I,2): PRINT #2, AFLD(I,3)
         PRINT #2, AFLD(I,4): PRINT #2, SAFLD(I,1)
         PRINT #2, SAFLD(I,2)
      NEXT I
      FOR I = 1 TO MAXWF
          PRINT #2, WF(I, 1): PRINT #2, WF(I, 2): PRINT #2, WF(I, 3)
      NEXT I
      FOR I = 1 TO MAXRTES
          FOR J = 1 TO MAXBSE
             PRINT #2, ROUTES(I, J)
          NEXT J
      NEXT I
      PRINT #2, NUMMSNS, NUMTEMPS
```

```
FOR I = 1 TO MAXMSNS + MAXPLANS
    FOR J = 1 TO IMSTTOP
       PRINT #2, IMST(I, J)
    NEXT J
NEXT I
PRINT #2, PKEY
FOR I = 1 TO MAXLEGS
    FOR J = 1 TO 7
       PRINT #2, LEGS(I, J)
    NEXT J
NEXT I
PRINT #2, SDTE
FOR I = 1 TO NUMACFT
    PRINT #2, ACFT(I, 1), ACFT(I, 2), ACFT(I, 3)
    PRINT #2, SJETS(I)
    FOR J = 1 TO 6: PRINT #2, GNDTIMES(I, J): NEXT J
NEXT I
' EXTRA VARIABLE FILE
PRINT #2, JLAT, JDLA, JTC, JCB, JFI, JDOC, LEAP, LSTLEAP
FOR I = 1 TO MAXPLANS
    PRINT #2, SNOTE(I)
NEXT I
FOR I = 1 TO MAXMSNS + MAXPLANS
    PRINT #2, SMST(I)
NEXT I
FOR I = 1 TO MAXRTES
    FOR J = 1 TO MAXHOPS
       PRINT #2, DIST(I, J, 1)
       PRINT #2, DIST(I, J, 2)
    NEXT J
NEXT I
FOR I = 1 TO MAXRTES
    FOR J = 1 TO MAXHOPS
       FOR K = 1 TO NUMACFT
          PRINT #2, FLTTIME(I, J, K)
       NEXT K
    NEXT J
NEXT I
PRINT #2, NSQDEP, ALPHA, OMEGA
FOR I = 1 TO MAXPLANS
    PRINT #2, NSQDPLY(I, 1), NSQDPLY(I, 2)
NEXT I
CLOSE #2
CLS
RETURN
'
'
```

```
11020 FOR I = CTR TO CTR + 19
          IF CTR > MAXWF THEN 11040
          IF WF(I, 1) = 0 THEN I = CTR + 19: GOTO 11040
          PRINT I; " "; SAFLD(WF(I, 1), 2); "  TO  ";
          PRINT SAFLD(WF(I, 2), 2); "  =  "; WF(I, 3)
11040 NEXT I
11050 LOCATE 22, 10: PRINT SPC(50);
      LOCATE 22, 10
      INPUT "ENTER NUMBER TO CHANGE,
             'N' FOR NEXT PAGE, 'ENTER' TO EXIT"; SWF
      IF SWF = "N" OR SWF = "n" THEN CLS : CTR = I - 1: GOTO 11020
      IF SWF = "" THEN RETURN
      Y = VAL(SWF) + 1
      LOCATE Y, 25
      INPUT WF(Y - 1, 3)
      '
      GOTO 11050
      '
      '  main menu
      '
20510 CLS 'SPC(24); STRING$(27, "_"): PRINT
      IF IMST(MAXMSNS + 1, 1) = 0 THEN
         LOCATE 8, 31: PRINT "NO MISSION TEMPLATES."
         LOCATE 10,19
        PRINT "YOU MUST COMPLETE INPUT CHECKLIST (item 5)"
         LOCATE 12,20
        PRINT"OR LOAD A FLOW PLAN FROM DISK (main menu)."
         LOCATE 14, 33: INPUT "ENTER TO RETURN"; SS: RETURN
      END IF
      PRINT TAB(34); "FLOW MENU": PRINT SPC(24); STRING$(27, "_"): PRINT
      PRINT TAB(25); "1 "; SDOT10; " FLOW & MOG": PRINT
      PRINT TAB(25); "2 "; SDOT10; " SLIDE MISSIONS": PRINT
      PRINT TAB(25); "3 "; SDOT10; " CHANGE MISSIONS"
      PRINT SPC(24); STRING$(27, "_"): PRINT
      PRINT SPC(34); "OUTPUT": PRINT
      PRINT TAB(25); "4 "; SDOT10; " MISSION SCHEDULE": PRINT
      PRINT TAB(25); "5 "; SDOT10; " FUEL REQUIRED": PRINT
      PRINT TAB(25); "6 "; SDOT10; " FORCE CLOSURE"
      PRINT SPC(24); STRING$(27, "_"): PRINT
      PRINT TAB(25); "7"; SDOT10; SDOT10; "MAIN MENU": PRINT
      PRINT : PRINT SPC(30);
      INPUT "ENTER CHOICE  "; IANSW: CLS
      IF IANSW < 1 OR IANSW > 7 THEN 20510
20640 IF IANSW = 7 THEN RETURN
20650 ON IANSW GOSUB 40000,23170,20780,29110,20700,20750, 20650
      GOTO 20510
      '
```

```
20700 ' compute fuel required
      '
      IF IMST(1, 1) = 0 THEN
          LOCATE 10, 20
          INPUT "NO FLOW PLAN.  'ENTER' TO RETURN . . ."; SS
          RETURN
      END IF
      '
      FUEL = 0
      TOTHRS = 0
      FOR I = 1 TO AFLDS
          FUEL(I) = 0
      NEXT I
      '
      FOR I = 1 TO (NUMMSNS * 5)
          IF LEGS(I, 3) <> 0 THEN
            FUELFLOW = ACFT(IMST(LEGS(I, 1), 1), 3)
            HRS = (LEGS(I, 4) / 60)
            TOTHRS = TOTHRS + HRS
            FUEL(LEGS(I, 3)) = FUEL(LEGS(I, 3)) + (FUELFLOW * HRS)
            FUEL = FUEL + (FUELFLOW * HRS)
          END IF
      NEXT I
      '
      CLS : CTR = 0
      PRINT "FIGURES REPRESENT THOUSANDS OF POUNDS OF FUEL REQUIRED"
      PRINT "REQUIRED TO SUPPORT THIS DEPLOYMENT PLAN." :PRINT
      PRINT "CHOKE POINT -> "; SAFLD(JCB, 2); " <-": PRINT
      PRINT "          BASE                FUEL (THOUSAND POUNDS)"
      PRINT "          ----                ----"
      FOR I = 1 TO AFLDS
          IF FUEL(I) <> 0 THEN
            CTR = CTR + 1
            PRINT "          "; SAFLD(I, 2),
            PRINT RIGHT$("        " + STR$(INT(FUEL(I))), 6)
            IF CTR MOD 15 = 0 THEN
              LOCATE 23, 1: INPUT "ENTER TO CONTINUE"; IANSW
              LOCATE 23, 1: PRINT SPC(30); ""
            END IF
          END IF
      NEXT I
      PRINT
      PRINT "          TOTAL               "; INT(FUEL): PRINT
      PRINT "          FLYING HOURS    "; INT(TOTHRS): PRINT
      PRINT : INPUT "ENTER TO CONTINUE"; IANSW
      RETURN
```

```
'
' deployment closure
'
20750 IF NSQDEP = 0 THEN
        LOCATE 10, 20
         INPUT "NO FLOW PLAN.  'ENTER' TO RETURN . . ."; SS
         RETURN
      END IF
      '
      CTR = 1: PTR = 1
      REDIM SCLOSE(NSQDEP, 5)
      FOR I = 1 TO NSQDEP
          IZTME = IMST(PTR, ICTM) + LEGS(IMST(PTR, IFSTLG), 2)
          GOSUB 29600
          SCLOSE(I,1)=RIGHT$("    "+STR$(ISQUAD(IMST(PTR, ISQD),1)),3)
          SCLOSE(I, 2) = SAFLD(INT(IMST(PTR, IFSTBSE) / 100), 2)
          SCLOSE(I, 3) = STME
          PTR = PTR + NSQDPLY(I, 2) - 1
          FOR J = ILSTLG TO IFSTLG STEP -1
             IF IMST(PTR, J) <> 0 THEN
               IZTEMP = LEGS(IMST(PTR, J), 4)
               IZTME = IMST(PTR, ICTM) + LEGS(IMST(PTR, J), 2) + IZTEMP
               GOSUB 29600
               ITEMP = INT(IMST(PTR, J + IFSTBSE - IFSTLG + 1) / 100)
               SCLOSE(I, 4) = SAFLD(ITEMP, 2)
               SCLOSE(I, 5) = STME
               EXIT FOR
             END IF
          NEXT J
          PTR = PTR + 1
      NEXT I
      '
      CLS : PRINT "CHOKE POINT -> "; SAFLD(JCB, 2); " <-": PRINT
      PRINT "SQD     DEPT    FIRST TAKEOFF    CLOSE"
      PRINT "---     ----    -------------    -----"
      FOR I = 1 TO NSQDEP
          FOR J = 1 TO 5
             PRINT SCLOSE(I, J); "     ";
          NEXT J
          PRINT
      NEXT I
      PRINT : INPUT "'ENTER' TO CONTINUE  "; IANSW
      RETURN
      '
```

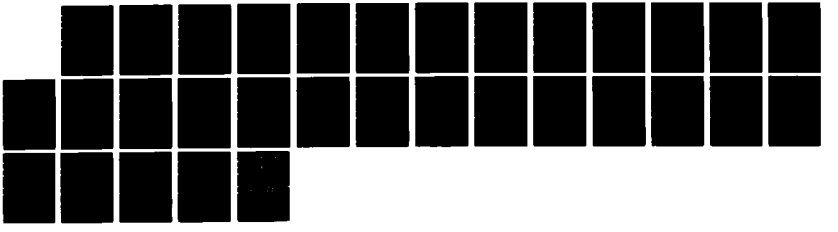
```
      '  changes menu
      '  called from 20650
      '
20780 CLS
      IF IMST(1, 1) = 0 THEN
         LOCATE 10, 15
         INPUT 'NO FLOW PLAN.  'ENTER' TO RETURN TO FLOW MENU  '; SS
         RETURN
      END IF
      '
      PRINT : PRINT
      PRINT TAB(30); 'CHANGE MISSIONS': PRINT : PRINT
      PRINT TAB(19); '1'; SDOT10; 'MISSION NUMBERS': PRINT
      PRINT TAB(19); '2'; SDOT10; 'CHANGE GROUND TIMES     (REFLOW)'
      PRINT
      PRINT TAB(19); '3'; SDOT10; 'CHANGE FLIGHT TIMES     (REFLOW)'
      PRINT
      PRINT TAB(19); '4'; SDOT10; 'REPOSITION IN THE FLOW  (REFLOW)'
      PRINT
      PRINT TAB(19); '5'; SDOT10; 'LINK / UNLINK MISSIONS': PRINT
      PRINT TAB(19); '6'; SDOT10; SDOT10; 'RETURN TO FLOW MENU':PRINT
      LOCATE 19, 30: INPUT 'ENTER CHOICE  '; ICGCD
      CLS : IF ICGCD > 6 THEN BEEP: GOTO 20780
      IF ICGCD = 6 THEN RETURN
      ON ICGCD GOSUB 26610, 23600, 23870, 24080, 27320, 20780
      GOTO 20780
      '
      '  flow and mog
      '  called from 23760,23980,24460,24710,25290,25440, 26150,26210
      '
21010 FLOWFLAG = 1 ' for mission numbers
      PKEY = 0
      FOR I = 1 TO NUMMSNS
          J = IFSTBSE: PJMP = 0: PCBFG = 0
         PEBFG = 0: PACCUM = 0: LKFLG = 0
          IF IMST(I, ILKFRM) <> 0 OR IMST(I, ILKTO) <> 0 THEN LKFLG = 1
          IF LKFLG = 0 THEN IMST(I, IPOST) = 0
          IMST(I, IPRE) = 0: IMST(I, ICBGND) = 0
          PKEY = (I - 1) * 5
21090     PKEY = PKEY + 1
          LEGS(PKEY, 1) = I: IMST(I, J - 5) = PKEY: PDSEQ = IMST(I, J)
          PTYP = IMST(I, IJET): PDBSE = INT(PDSEQ / 100)
          '
```

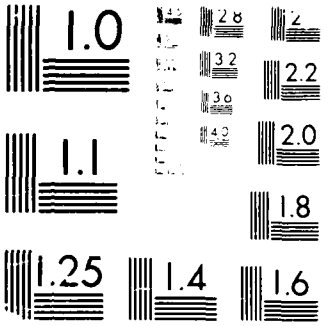
```
            '   has mission passed choke point ?
            '
            IF PDBSE <> JCB THEN 21190
            PCBFG = PCBFG + 1
            IF PCBFG > 1 THEN 21190
            PDFCT = PDSEQ - (PDBSE * 100)
            IMST(I, ICBGND) = GNDTIMES(PTYP, PDFCT)
21190       LEGS(PKEY, 3) = PDBSE: PASEQ = IMST(I, J + 1)
            PABSE = INT(PASEQ / 100)
            LEGS(PKEY, 6) = PABSE
            PAFCT = PASEQ - PABSE * 100: LEGS(PKEY, 7) = PAFCT
            IF LEGS(PKEY,5)=0 THEN LEGS(PKEY,5)=GNDTIMES(PTYP, PAFCT)
            FOR II = 1 TO MAXRTES
               FOR JJ = 1 TO MAXHOPS
                  IF ROUTES(II,JJ)=PDBSE AND ROUTES(II, JJ+1)= PABSE THEN
                     IF LEGS(PKEY, 4) = 0 THEN
                       LEGS(PKEY, 4) = FLTTIME(II, JJ, PTYP)
                     END IF
                     II = MAXRTES
                     EXIT FOR
                  END IF
               NEXT JJ
            NEXT II
            IF PCBFG >= 1 THEN 21290
            '
            '   store pre-choke point time
            '
            PRETM = LEGS(PKEY,4) + LEGS(PKEY,5): LEGS(PKEY,2) = 0 - PRETM
            IMST(I, IPRE) = IMST(I, IPRE) + LEGS(PKEY, 2): GOTO 21500
21290       IF PDBSE <> JCB OR PCBFG > 1 THEN 21430
            '
            '   compute mission start time
            '
            PTMP1 = LEGS(PKEY - PJMP,2):LEGS(PKEY - PJMP,2)= IMST(I,IPRE)
21340       PJMP = PJMP - 1
            IF PJMP <= 0 THEN 21380
            PTMP2 = LEGS(PKEY - PJMP, 2)
            LEGS(PKEY - PJMP, 2) = IMST(I, IPRE) - PTMP1
            PTMP1 = PTMP1 + PTMP2: GOTO 21340
21380       LEGS(PKEY, 2) = 0
            IF LKFLG=0 THEN IMST(I,IPOST)=IMST(I,IPOST)+LEGS( PKEY,4)
            PACCUM = PACCUM + LEGS(PKEY, 4) + LEGS(PKEY, 5)
            IF LKFLG=0 THEN IMST(I,IPOST)=IMST(I,IPOST)+LEGS( PKEY,5)
            GOTO 21500
21430       LEGS(PKEY, 2) = PACCUM
            PACCUM = PACCUM + LEGS(PKEY, 4) + LEGS(PKEY, 5)
            IF PEBFG = 1 THEN 21500
            IF LKFLG=0 THEN IMST(I,IPOST)=IMST(I,IPOST)+ LEGS(PKEY,4)
            '
```

```
          '   store choke point-beddown base time
          '
          IF LKFLG=0 THEN IMST(I,IPOST)=IMST(I,IPOST)+ LEGS(PKEY,5)
21500     J = J + 1: PJMP = PJMP + 1
          IF J <> ILSTBSE THEN IF PAFCT <> 6 THEN 21090
          '
          '   end of mission sequence
          '
          IF IMST(I, ICTM) = 0 THEN 21560
          IF ISLFLG = 1 THEN 21570
21560     IF LKFLG = 0 THEN IMST(I, ICTM) = JLAT
21570 NEXT I
      '
      ' smooth flow missions from choke point
      '
      IF ISLFLG = 1 THEN 22320
      FOR I = 2 TO NUMMSNS
          IF IMST(I, ILKFRM) <> 0 THEN GOSUB 23030: GOTO 21650
          IMST(I, ICTM) = IMST(I - 1, ICTM) + JFI
21650 NEXT I
      '
      ' adjust flow for mog  Programmed by Capt Dean Farwell
      '                          for Fragbuster
      '
      NMOG = AFLD(JCB, 3): WMOG = AFLD(JCB, 4)
      FOR K = 1 TO 9
          JARAY(K, 1) = 0: JARAY(K, 2) = 0
         NARAY(K, 1) = 0: NARAY(K, 2) = 0
      NEXT K
      '
      IPAD = IMST(1, ICTM) - JLAT
      IF IPAD <= 0 THEN 21810
      '
      FOR K = 1 TO NUMMSNS
          IMST(K, ICTM) = IMST(K, ICTM) - IPAD
      NEXT K
```

AD-A190 680 DEPLOY: AN AIRLIFT DEPLOYMENT SCHEDULER FOR REAL-TIME CRISIS ACTION PLANNING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI R M FOSTER 2/2

UNCLASSIFIED MAR 88 AFIT/GST/ENS/88M-4 F/G 15/5 NL





```
21810 NCTR = 0: JCTR = 0
      FOR I = ALPHA TO OMEGA
          IF ACFT(IMST(I, IJET), 2) = 1 THEN 22030
          '
          IF IMST(I, ICTM) - IMST(I, ICBGND) < NARAY(1, 2) THEN 21900
          GOSUB 22750: GOTO 22240
          '
          '    slide missions
          '
21900     IF NCTR = NMOG THEN 21920
          GOSUB 22630: GOTO 22240
          '
21920     IF IMST(I, ICTM) - IMST(I, ICBGND) < JARAY(1  2) THEN 21940
          GOSUB 22810: GOTO 22240
          '
21940     '
          IF JCTR = WMOG THEN 21980
          GOSUB 22690: GOTO 22240
          '
21980     IF NARAY(1, 2) > JARAY(1, 2) THEN 22010
          GOSUB 22870: GOSUB 22750: GOSUB 22600: GOTO 22240
          '
22010     GOSUB 22940: GOSUB 22810: GOSUB 22600: GOTO 22240
          '
22030     IF IMST(I, ICTM) - IMST(I, ICBGND) < JARAY(1, 2) THE.. 22060
          GOSUB 22810: GOTO 22240
          '
22060     IF JCTR = WMOG THEN 22090
          GOSUB 22690: GOTO 22240
        '
22090     K = 0
          FOR J = 1 TO JCTR
             IF ACFT(IMST(JARAY(J, 1), IJET), 2) = 1 THEN 22130
             K = J: J = JCTR
22130     NEXT J
          IF K <> 0 THEN 22170
          GOSUB 22940: GOSUB 22810: GOSUB 22600: GOTO 22240
          '
22170     IF NARAY(1, 2) < JARAY(1, 2) THEN 22200
          GOSUB 22940: GOSUB 22810: GOSUB 22600: GOTO 22240
          '
22200     GOSUB 22870
          NARAY(1, 1) = JARAY(K, 1): NARAY(1, 2) = JARAY(K, 2)
          JARAY(K, 1) = I: JARAY(K, 2) = IMST(I, ICTM)
          GOSUB 22380: GOSUB 22500
22240 NEXT I
      '
      GOSUB 27100    ' mission number sequencing
22320 RETURN 'TO  23760,23980,24460,24710,25290,25440,26150, 26320
      '
```

```
      ' mogger subroutines
      '
      ' bubble sort JARAY
      '
22380 NN = 0
      FOR N = 2 TO JCTR
          IF JARAY(N - 1, 2) <= JARAY(N, 2) THEN 22440
          SWAP JARAY(N - 1, 1), JARAY(N, 1)
          SWAP JARAY(N - 1, 2), JARAY(N, 2)
          NN = 1
22440 NEXT N
      IF NN <> 0 THEN 22380
      RETURN
      '
      ' bubble sort NARAY
      '
22500 NN = 0
      FOR N = 2 TO NCTR
          IF NARAY(N - 1, 2) <= NARAY(N, 2) THEN 22560
          SWAP NARAY(N - 1, 2), NARAY(N, 2)
          SWAP NARAY(N - 1, 1), NARAY(N, 1)
          NN = 1
22560 NEXT N
      IF NN <> 0 THEN 22500
      RETURN
      '
22600 IF MOGCKFLG = 1 THEN
         LPRINT "MOG VIOLATION AT MISSION SEQUENCE # "; I: RETURN
      END IF
      FOR N = I + 1 TO NUMMSNS
          IMST(N, ICTM) = IMST(N, ICTM) + MADD
      NEXT N
      RETURN
      '
      ' put into NARAY
      '
22630 NCTR = NCTR + 1
      NARAY(NCTR, 1) = I: NARAY(NCTR, 2) = IMST(I, ICTM)
      GOSUB 22500: RETURN
      '
      '  put into JARAY
      '
22690 JCTR = JCTR + 1
      JARAY(JCTR, 1) = I: JARAY(JCTR, 2) = IMST(I, ICTM)
      GOSUB 22380: RETURN
      '
      ' put in top of NARAY
      '
22750 IF NCTR = 0 THEN NCTR = 1
      NARAY(1, 1) = I: NARAY(1, 2) = IMST(I, ICTM)
      GOSUB 22500: RETURN
      '
```

```
22810 ' put in top of JARAY
      '
      IF JCTR = 0 THEN JCTR = 1
      JARAY(1, 1) = I: JARAY(1, 2) = IMST(I, ICTM)
      GOSUB 22380: RETURN
      '
22870 ' calculate slide for NARAY
      '
      IF MOGCKFLG = 1 THEN 22920
      MADD = NARAY(1, 2) - (IMST(I, ICTM) - IMST(I, ICBGND))
      IMST(I, ICTM) = NARAY(1, 2) + IMST(I, ICBGND)
22920 RETURN
      '
22940 ' calculate slide for JARAY
      '
      IF MOGCHFLG = 1 THEN 22990
      MADD = JARAY(1, 2) - (IMST(I, ICTM) - IMST(I, ICBGND))
      IMST(I, ICTM) = JARAY(1, 2) + IMST(I, ICBGND)
22990 RETURN
      '
      '  called from 21630
      '
23030 FOR IK = ILSTLG TO IFSTLG STEP -1
          IF IMST(IMST(I, ILKFRM), IK) <> 0 THEN LSTSTP = IK: IK=IFSTLG
      NEXT IK
      CTR1 = IMST(IMST(I, ILKFRM), ICTM)
      CTR2 = IMST(I, ILKFRM)
      CTR2A = LEGS(IMST(CTR2, LSTSTP), 2)
      CTR3 = IMST(I, ILKFRM)
      CTR3A = LEGS(IMST(CTR3, LSTSTP), 4)
      CTR4 = IMST(I, ILKFRM)
      CTR4A = LEGS(IMST(CTR4, LSTSTP), 5)
      CTR5 = LEGS(IMST(I, IFSTLG), 2)
      IMST(I, ICTM) = CTR1 + CTR2A + CTR3A + CTR4A - CTR5
23150 RETURN     '  to 21630
      '
      ' slide missions
      '     called from  20650
      '
23170 CLS
      IF IMST(1, 1) = 0 THEN
         LOCATE 10, 15
         INPUT 'NO FLOW PLAN.  'ENTER' TO RETURN TO FLOW MENU  '; SS
         RETURN
      END IF
      LOCATE 10, 15: PRINT 'DO YOU WANT TO SLIDE ONE MISSION     (1)'
      LOCATE 12, 15: PRINT 'A CONTINOUS SERIES OF MISSIONS       (2)'
      LOCATE 14, 15
      INPUT 'OR RETURN TO MAIN MENU                (3)  '; IANSW: CLS
      IF IANSW > 3 OR IANSW < 1 THEN BEEP: GOTO 23170
      ON IANSW GOTO 23290, 23490, 23250
23250 RETURN
```

```
'
'   slide one mission
'   called from 23240
'
23290 ISLDFLG = 1: GOSUB 23340
      IF IMST(IS1, ILKFRM) <> 0 THEN
         BEEP:PRINT "MSN "; IS1; " LINKED TO MSN "; IMST(IS1, ILKFRM)
         INPUT "ENTER TO CONTINUE"; SANSW: GOTO 23450
      END IF
      IF IMST(IS1, ILKTO) <> 0 THEN
         BEEP:PRINT "MSN "; IS1; " LINKED TO MSN "; IMST(IS1, ILKTO)
         INPUT "ENTER TO CONTINUE"; SANSW: GOTO 23450
      END IF
      IMST(IS1, ICTM) = NUTIM: GOTO 23170
      '
      '    called from 23290,23490
      '
23340 LOCATE 10, 15
      INPUT "ENTER SEQUENCE # OF FIRST MISSION TO BE MOVED  "; IS1
      IF IS1 > MAXMSNS OR IS1 < 1 THEN BEEP: GOTO 23340
      '
23360 LOCATE 12, 15
      INPUT"ENTER (DDMMYY) OF NEW TAKEOFF FROM CHOKE POINT"; SDATE
      IF LEN(SDATE) <> 6 THEN
          BEEP : LOCATE 12,63 : PRINT SPC(15); :GOTO 23360
      END IF
      GOSUB 29850       ' convert calendar to julian
      IF BADFLG = 1 THEN
          LOCATE 12, 63: PRINT SPC(15); : GOTO 23360
      END IF
      NDY = JULIAN
      '
23400 LOCATE 14, 15
      INPUT "ENTER NEW TAKEOFF TIME (HHMM) FROM CHOKE POINT"; SNTM
      IF LEN(SNTM) <> 4 THEN
          BEEP: LOCATE 14, 63: PRINT SPC(20); : GOTO 23400
      END IF
      GOSUB 24790     ' time conversion days -> min
      IF BADFLG = 1 THEN
         LOCATE 14, 63: PRINT SPC(20); : GOTO 23400
      END IF
      LOCATE 23, 22: INPUT "IS THIS INFO CORRECT (Y/N) "; SANSW
      IF SANSW <> "Y" THEN CLS : GOTO 23340
      '
23450 RETURN      '  to 23290, 23490
      '
```

```
      '  slide a series of missions
      '  called from 23240
23490 ISLDFLG = 1: GOSUB 23340
      IDEC = NUTIM - IMST(IS1, ICTM)
      FOR K = IS1 TO NUMMSNS
          IF IMST(K, ILKFRM) <> 0 THEN
            IF IMST(K, ILKFRM) < IS1 THEN
              BEEP: PRINT K; " LINKED TO "; IMST(K, ILKFRM)
              INPUT "ENTER TO CONTINUE  "; SANSW: GOTO 23540
            END IF
          END IF
          IMST(K, ICTM) = IMST(K, ICTM) + IDEC
23540 NEXT K
      GOTO 23150
      '
      '  change ground times
      '  called from changes menu
      '
23600 CLS : LOCATE 10, 20
      INPUT "HOW MANY LEGS TO CHANGE  "; NGTLG: PRINT : PRINT
      FOR I = 1 TO NGTLG
          CLS : LOCATE 10, 2
          PRINT "REFER TO PRINTED MISSION SCHEDULE FOR LEG NUMBERS"
23650     LOCATE 12,2:PRINT"LEG NUMBER CONTAINING THE ARRIVAL BASE";
          INPUT"WHOSE GROUND TIME WILL CHANGE  "; LEGN:PRINT :PRINT
          IF LEGN > MAXLEGS OR LEGN < 1 THEN BEEP: GOTO 23650
          IF IMST(LEGS(LEGN, 1), ILKTO) <> 0 THEN GOSUB 23790 ELSE 23720
          IF IMST(LEGS(LEGN, 1), LSTSTP) <> LEGN THEN 23720
          PRINT "LINKED LEG . . .";
          INPUT "DO YOU STILL WANT TO CHANGE GND TIME (Y/N)  "; SANSW
          IF SANSW <> "Y" THEN 23750
23720     PRINT "NEW GROUND TIME AT ";
          PRINT SAFLD(LEGS(LEGN, 6), 1); " (HHMM)  ": INPUT SNTM
          IF LEN(SNTM) <> 4 THEN BEEP: GOTO 23720
          NDAY = 0: GOSUB 24830: LEGS(LEGN, 5) = NUTIM
23750 NEXT I
      GOSUB 21010: RETURN     ' to 900
      '
      '  called from 23680
23790 FOR IK = ILSTLG TO IFSTLG STEP -1
          IF IMST(LEGS(LEGN, 1), IK) <> 0 THEN LSTSTP = IK: IK = IFSTLG
      NEXT IK
      RETURN       '  to 23680
      '
```

```
      '  change flight times
      '  called from changes menu
      '
23870 CLS
      LOCATE 10, 20: INPUT "HOW MANY LEGS TO CHANGE  "; NARLG
      Y = 10: X = 20
      FOR I = 1 TO NARLG
          CLS : LOCATE Y, X
23920     INPUT "LEG NUMBER YOU WANT TO CHANGE"; NLEGN: PRINT : PRINT
          IF NLEGN < 1 OR NLEGN > MAXLEGS THEN BEEP: GOTO 23920
23940     LOCATE Y+2,X: INPUT "FLY TIME FOR THIS LEG (HHMM)  "; SNTM
          IF LEN(SNTM) <> 4 THEN
            BEEP: LOCATE Y + 2, 49: PRINT SPC(10); : GOTO 23940
          END IF
          NDAY = 0: GOSUB 24830: LEGS(NLEGN, 4) = NUTIM
      NEXT I
      GOSUB 21010: RETURN   'TO 900
      '
      '   reposition a mission in the flow
      '   called from changes menu
      '
24080 LOCATE 10,20:PRINT "ENTER (1) TO SWAP TWO MISSIONS..."
      LOCATE 12,20
      PRINT "..... (2) TO REPOSITION A MISSION IN THE FLOW ..."
      LOCATE 14, 20: INPUT "OR... (3) TO RETURN TO MENU....."; IANSW
      IF IANSW < 1 OR IANSW > 3 THEN
         BEEP: LOCATE 14, 56: PRINT "      ": GOTO 24080
      END IF
      IF IANSW = 3 THEN RETURN      'to 900
      ON IANSW GOSUB 24190, 24520
      CLS : GOTO 24080
      '
```

```
      '  swap two missions
      '  called from 24130
      '
24190 CLS : LOCATE 10, 15
24200 INPUT "ENTER FLOW SEQUENCE NUMBERS TO SWAP (X,Y)": ISWP1, ISWP2
      IF IMST(ISWP1, ILKFRM) <> 0 THEN ILM = ISWP1: GOTO 24250
      IF IMST(ISWP1, ILKTO) <> 0 THEN ILM = ISWP1: GOTO 24250
      IF IMST(ISWP2, ILKFRM) <> 0 THEN ILM = ISWP2: GOTO 24250
      IF IMST(ISWP2, ILKTO) = 0 THEN 24270 ELSE ILM = ISWP2
24250 CLS : BEEP: LOCATE 10, 20
      PRINT ILM; " LINKED MISSION...CAN'T SWAP"
      LOCATE 12, 20: INPUT "ENTER TO CONTINUE......"; SANSW: GOTO 24080
24270 IF ISWP1 > NUMMSNS OR ISWP2 > NUMMSNS THEN 24290
      IF ISWP1 < 1 OR ISWP2 < 1 THEN 24290 ELSE 24300
24290 PRINT "BAD NUMBER...RETRY": BEEP: GOTO 24200
24300 IF ISWP1 > ISWP2 THEN 24310 ELSE 24350
24310 PRINT "WRONG ORDER...REENTER": BEEP: GOTO 24200
      '
      ' ***  24350 also called from 24690  ***
      '
24350 SWAP IMST(ISWP1, IJET), IMST(ISWP2, IJET)
      SWAP IMST(ISWP1, ISQD), IMST(ISWP2, ISQD)
      FOR LUPE = IFSTBSE TO ILSTBSE
          SWAP IMST(ISWP1, LUPE), IMST(ISWP2, LUPE)
      NEXT LUPE
      SWAP SMST(ISWP1), SMST(ISWP2)
      '
      GOSUB 50000
      '
      IF MOVFLG = 0 THEN GOSUB 21010     ' reflow
      RETURN     ' to 24130, 24690
      '
```

```
      '  move a mission
      '  called from 24130
      '
24520 CLS : MOVFLG = 0: LOCATE 10, 20
24530 PRINT "ENTER TWO SEQUENCE NUMBERS : ": PRINT : PRINT
      LOCATE 12, 20
      INPUT "MISSION TO BE PULLED , POSITION TO REINSERT"; NUM1, NUM2
      IF IMST(NUM1, ILKFRM) <> 0 THEN ILM = NUM1: GOTO 24600
      IF IMST(NUM1, ILKTO) <> 0 THEN ILM = NUM1: GOTO 24600
      IF IMST(NUM2, ILKFRM) <> 0 THEN ILM = NUM2: GOTO 24600
      IF IMST(NUM2, ILKTO) = 0 THEN 24620 ELSE ILM = NUM2
24600 CLS : BEEP: LOCATE 10, 20
      PRINT ILM; " LINKED MISSION...CAN'T SWAP"
      LOCATE 12, 20: INPUT "ENTER TO CONTINUE.....": SANSW: GOTO 24720
24620 IF NUM1 < 1 OR NUM2 < 1 THEN 24640
      IF NUM1 > NUMMSNS OR NUM2 > NUMMSNS THEN 24640 ELSE 24650
24640 CLS : PRINT "INVALID NUMBER": BEEP: GOTO 24530
24650 IF NUM2 > NUM1 THEN INC = 1 ELSE INC = -1
      FOR LUPE3 = NUM1 + INC TO NUM2 STEP INC
          ISWP1 = LUPE3 - INC
          ISWP2 = LUPE3
          MOVFLG = 1: GOSUB 24350
      NEXT LUPE3
      GOSUB 21010: MOVFLG = 0
24720 RETURN       'TO 4130
      '
      ' time conversion subroutines
      '
      ' convert days + hours + minutes to minutes
      ' called from 23440,23740,23960
      '
24790 BADFLG = 0: ITEMP = JDLA
      IF NDY - JDLA < 0 AND ABS(NDY - JDLA) > 21 THEN
          NDY = NDY + 365 + LSTLEAP: GOTO 24820
      END IF
      IF NDY - JDLA > 21 THEN ITEMP = JDLA + 365 + LSTLEAP
24820 NDAY = (NDY - ITEMP) * 1440
24830 NHR = VAL(LEFT$(SNTM, 2)) * 60  'from 23740,23960
      NMIN = VAL(RIGHT$(SNTM, 2))
      IF NHR < 0 OR NHR > 1440 OR NMIN < 0 OR NMIN > 59 THEN
          BADFLG = 1
      ELSE
          NUTIM = NMIN + NHR + NDAY
      END IF
      RETURN    'to 23440, 23740, 23960
```

```
      '
      '  mission numbers
      '  called from changes menu
      '
26610 CLS : Y = 8: X = 20
      LOCATE Y, X: PRINT "DO YOU WANT TO . . ."
      LOCATE Y + 2, X
      PRINT "CHANGE MISSION PREFIX FOR ONE MISSION       (1)"
      LOCATE Y + 4, X
      PRINT "OR CHANGE THE SEQUENCING OF MISSION NUMBERS (2)"
      LOCATE Y + 6, X
      INPUT "OR RETURN TO CHANGES TO MISSIONS MENU       (3)"; IANSW
      IF IANSW < 1 OR IANSW > 3 THEN 26610
26650 IF IANSW = 3 THEN RETURN  ' to 20900
      ON IANSW GOSUB 26680, 26800, 26650: GOTO 26610
      '
26680 CLS : LOCATE 12, 12
      INPUT "FLOW SEQUENCE NUMBER OF MISSION TO CHANGE"; IANSW
      IF IANSW < 1 OR IANSW > MAXMSNS + MAXPLANS THEN BEEP: GOTO 26680
      CLS : LOCATE 10, 20
      PRINT "YOU ARE CHANGING THE "; SMST(IANSW); " MISSION"
      LOCATE 14, 16
26730 INPUT "INPUT THE NEW MISSION PREFIX OR 'X' TO CANCEL"; SANSW
      IF LEFT$(SANSW, 1) = "X" THEN 26780
      IF LEN(SANSW) > 12 THEN BEEP: GOTO 26730
      SMST(IANSW) = LEFT$(SANSW + "************", 12)
      IF MID$(SANSW, 8, 1) <> "*" THEN NUMCHFLG = 1
26780 RETURN    ' to 26660
      '
26800 CLS : LOCATE 10, 20
      PRINT "DO YOU WISH TO RENUMBER BY AIRCRAFT TYPE (1) . . ."
      LOCATE 12, 20: PRINT "BY BASE OF ORIGIN (2) . . ."
26820 LOCATE 14, 20: INPUT "OR AS A CONTINUOUS FLOW (3)"; IANSW
      IF IANSW < 1 OR IANSW > 3 THEN BEEP: GOTO 26820
      ON IANSW GOSUB 26870, 27100, 27000
      RETURN    'TO 6660
      '
      '  from 6840
      '
26870 FOR I = 1 TO NUMACFT
          BCTR = 1
          CLS
          PRINT "NOW RENUMBERING ALL "; SJETS(I); " AIRCRAFT"
          INPUT "ENTER FIRST NUMBER IN SEQUENCE"; BCTR
          FOR J = 1 TO NUMMSNS
             IF IMST(J, IJET) <> I THEN 26960
          MID$(SMST(J), 8, 2) = RIGHT$(STR$(BCTR + 100), 2)
          BCTR = BCTR + 1
26960     NEXT J
      NEXT I
      RETURN  ' to 26840
      '
```

```
      '  called from 26840
      '
27000 CLS : LOCATE 10, 20
      INPUT "FIRST FLOW SEQUENCE # TO CHANGE"; NUMBR1: PRINT
      LOCATE 12, 20: INPUT "LAST FLOW SEQUENCE # TO CHANGE"; NUMBR2
      LOCATE 14, 20: PRINT "STARTING WITH FLOW SEQUENCE # "; NUMBR1
      LOCATE 14, 54: INPUT "ENTER FIRST #"; NUMBR3
      FOR I = NUMBR1 TO NUMBR2
         MID$(SMST(I), 8, 2) = RIGHT$(STR$(100 + NUMBR3), 2)
         NUMBR3 = NUMBR3 + 1
      NEXT I
      RETURN  ' TO 6840
      '
      ' from 6840
      '
27100 REDIM RTEMP(MAXRTES)
      FOR I = 1 TO MAXRTES
          TEMP = ROUTES(I, 1)   ' origin base
          IF TEMP = 0 THEN I = MAXRTES: EXIT FOR
          FOR J = 1 TO MAXRTES
             IF RTEMP(J) = TEMP THEN EXIT FOR
             IF RTEMP(J) = 0 THEN RTEMP(J) = TEMP: EXIT FOR
          NEXT J
      NEXT I
      '
      FOR I = 1 TO MAXRTES
          IF FLOWFLAG = 1 THEN PTR = 1: GOTO 27150
          IF RTEMP(I) = 0 THEN 27190
          CLS
         PRINT"FOR MISSIONS ORIGINATING AT";SAFLD(RTEMP(I), 2); " <"
          PRINT : INPUT "ENTER FIRST NUMBER IN SEQUENCE "; PTR
27150     FOR J = 1 TO NUMMSNS
             IF INT(IMST(J, IFSTBSE) / 100) <> RTEMP(I) THEN 27180
             STEMP1 = STR$(PTR + 100)
             MID$(SMST(J), 8, 2) = RIGHT$(STEMP1, 2)
             PTR = PTR + 1
27180     NEXT J
27190 NEXT I
      FLOWFLAG = 0
      RETURN  'to 26840
      '
```

```
      '  link missions
      '  called from changes menu
      '
27320 CLS
      LOCATE 10, 15: PRINT "DO YOU WANT TO LINK TWO MISSIONS   (1)... "
      LOCATE 12, 15: PRINT "... UNLINK TWO MISSIONS            (2)..."
      LOCATE 14, 15
      INPUT "...OR RETURN TO MAIN MENU          (3)"; IANSW: CLS
      IF IANSW > 3 OR IANSW < 1 THEN BEEP: GOTO 27320
      ON IANSW GOTO 27420, 27610, 27400
27400 RETURN      ' TO 900
      '
27420 PRINT "                          'ENTER' TO RETURN TO MENU":PRINT
27430 INPUT "ENTER SEQUENCE # OF FIRST MISSION"; ILPR1: PRINT: PRINT
      IF ILPR1 = 0 THEN 27320
      INPUT "ENTER SEQUENCE # OF SECOND MISSION"; ILPR2: PRINT: PRINT
      IF ILPR2 = 0 THEN 27320
      IF ILPR1 >= ILPR2 THEN BEEP: PRINT "WRONG ORDER": GOTO 27430
      GOSUB 27780
      STEMP1 = SAFLD(LEGS(IMST(ILPR1, LSTSTP), 6), 1)
      STEMP2 = SAFLD(LEGS(IMST(ILPR2, IFSTLG), 3), 1)
      IF STEMP1 <> STEMP2 THEN BEEP: PRINT "BASE MISMATCH": GOTO 27430
      IF IMST(ILPR1, IJET) <> IMST(ILPR2, IJET) THEN
         BEEP: PRINT "JET MISMATCH": GOTO 27430
      END IF
      IMST(ILPR1, ILKTO) = ILPR2: IMST(ILPR2, ILKFRM) = ILPR1
      GOSUB 27780
27520 PRINT "ENTER GROUND TIME AT ";
      PRINT SAFLD(LEGS(IMST(ILPR1, LSTSTP), 6), 2); " (HHMM);"
      INPUT ZTME: IF LEN(ZTME) <> 4 THEN 27520
      GOSUB 27880: NEGT = JTME
      ITEMP1=LEGS(IMST(ILPR1,LSTSTP),2) + LEGS(IMST(ILPR1, LSTSTP), 4)
      ITEMP2 = NEGT - LEGS(IMST(ILPR2, IFSTLG), 2)
      IMST(ILPR2, ICTM) = IMST(ILPR1, ICTM) + ITEMP1 + ITEMP2
      LEGS(IMST(ILPR1, LSTSTP), 5) = NEGT
      GOTO 27320
```

```
'
'  unlink missions
'
27610 CLS : PRINT SPC(20); ''ENTER' TO RETURN TO MENU': PRINT
27620 INPUT 'ENTER SEQUENCE # OF FIRST MISSION'; ILPR1: PRINT:PRINT
      IF ILPR1 = 0 THEN 27320
      INPUT 'ENTER SEQUENCE # OF SECOND MISSION'; ILPR2
      IF ILPR2 = 0 THEN 27320
      IF IMST(ILPR1,ILKTO)=0 THEN IF IMST(ILPR2,ILKFRM) = 0 THEN 27740
      IF ILPR1 >= ILPR2 THEN BEEP: PRINT 'WRONG ORDER': GOTO 27620
      IF IMST(ILPR1,ILKTO) <> ILPR2 THEN
        BEEP:PRINT'MISMATCH': GOTO 27620
      END IF
      IF IMST(ILPR2,ILKFRM)<>ILPR1 THEN
        BEEP:PRINT 'MISMATCH': GOTO 27620
      END IF
      GOSUB 27780
      LEGS(IMST(ILPR1, LSTSTP), 5) = 0
      IMST(ILPR1, ILKTO) = 0: IMST(ILPR2, ILKFRM) = 0
27730 RETURN
27740 INPUT 'NEITHER MISSION LINKED. ENTER TO CONTINUE'; SANSW
      GOTO 27730
'
' called from 27480, 27700, 27510
'
27780 FOR M = ILSTLG TO IFSTLG STEP -1
          IF IMST(ILPR1, M) <> 0 THEN LSTSTP = M: M = IFSTLG
      NEXT M
      RETURN
'
' convert hrs+min to minutes
' called from 27540
'
27880 IMIN = VAL(RIGHT$(ZTME, 2))
      IHR = VAL(LEFT$(ZTME, 2)) * 60
      JTME = IMIN + IHR
      RETURN
'
' called from 5260,5820,6140,6400
'
27950 FOR IZ = 1 TO 5
          LEGS((IANSW - 1) * 5 + IZ, 4) = 0
          LEGS((IANSW - 1) * 5 + IZ, 5) = 0
      NEXT IZ
      RETURN
```

```
      '
      '  screen schedule
      '  called from 29120
      '
28610 CLS
      ' first ckoke point takeoff
      IZTME = JLAT: GOSUB 29600
      PRINT : PRINT : PRINT
      Y = VAL(MID$(SDTE, 3, 2)) * 3 - 2
      STME = LEFT$(SDTE, 2) + ' ' + MID$(ZMONTH, Y, 3)
      STME = STME + ' ' + RIGHT$(SDTE, 2) + ' ' + T$
      PRINT 'FIRST DEPARTURE TIME    >  '; STME; ' AT '; SAFLD(JCB,2)
      PRINT 'DEPARTURE INTERVAL      >  '; JFI; ' MINUTES'
      PRINT 'NUMBER OF MISSIONS      >  '; NUMMSNS: PRINT
28680 INPUT 'ENTER SEQUENCE NUMBER OF FIRST MISSION TO PRINT ';JFIRST
      PRINT : IF JFIRST < 1 OR JFIRST > NUMMSNS THEN BEEP: GOTO 28680
28700 INPUT 'ENTER FLOW SEQUENCE NUMBER OF LAST MISSION TO PRINT';JLAST
      PRINT
      IF JLAST < JFIRST OR JLAST > NUMMSNS THEN BEEP: GOTO 28700
```

```
      CLS
      FOR I = JFIRST TO JLAST
         PRINT
          IZTME = IMST(I, ICTM): GOSUB 29600: S1 = STME
          IXTME = IMST(I, ICTM) + LEGS(IMST(I, IFSTLG), 2)
          IZTME = IXTME: GOSUB 29600' = INT(IXTME / 1440)
          MID$(SMST(I), 10, 3) = RIGHT$(STR$(JULIAN + 1000), 3)
28820     '
28840     PRINT 'FLOW SEQUENCE #  '; I; '     '; SAFLD(JCB, 2);
          PRINT ' DEPARTURE TIME '; S1
           PRINT 'ACFT '; SJETS(IMST(I, IJET)); '       ';
           PRINT ISQUAD(IMST(I, ISQD), 1); ' TAS ';
           PRINT '      MISSION #  '; SMST(I): PRINT
           PRINT 'ARRIVE      GNDTM         DEPART       FLTTM   LEG'
         '
           FOR J = IFSTLG TO ILSTLG
               IF J <> IFSTLG THEN ZFRM = '': GOTO 28920
               IF IMST(I, ILKFRM) <> 0 THEN
                 ZFRM = ' CYCLES FROM ' + SMST(IMST(I, ILKFRM))
               ELSE
                 ZFRM = ''
               END IF
               IF IMST(I, ILKTO) <> 0 THEN
                 ZTO = ' CYCLES TO ' + SMST(IMST(I, ILKTO))
               ELSE
                 ZTO = ''
               END IF
28920         IF IMST(I, J) = 0 THEN J = ILSTLG: GOTO 29020
               '
               ITMP2 = LEGS(IMST(I, J), 4): GOSUB 29770
               IZTME = IMST(I, ICTM) + LEGS(IMST(I, J), 2): GOSUB 29600
               IF J = IFSTLG THEN PRINT SPC(23);
               PRINT SAFLD(LEGS(IMST(I, J), 3), 2); '     '; STME; '  ';
               PRINT STME2; '     '; IMST(I, J)
               ITMP2 = LEGS(IMST(I, J), 5): GOSUB 29770
               IZTME = IZTME + LEGS(IMST(I, J), 4): GOSUB 29600
               PRINT STME + '  ' + STME2 + '     ';
               IF IMST(I, J + 1) = 0 OR J = ILSTLG THEN
                 PRINT ' '; SAFLD(LEGS(IMST(I, J), 6), 2)
               END IF
29020     NEXT J
           PRINT ZTO', LEAP, LSTLEAP
           INPUT '                    HIT RETURN TO CONTINUE  '; SANSW
      NEXT I
      RETURN        ' to 650
```

```
      '
      '  printed mission schedule
      '  called from flow menu
      '
29110 CLS
      IF NUMMSNS = 0 THEN
         LOCATE 8, 31
         PRINT "NO FLOW PLAN."
         PRINT : PRINT SPC(22); "YOU MUST RETURN TO FLOW MENU"
        PRINT : PRINT SPC(17); "AND SELECT MISSIONS TO FLOW (OPTION 1)."
         PRINT : PRINT SPC(25); "HIT 'ENTER' TO CONTINUE"
         LOCATE CSRLIN - 1, 55: INPUT SS
         RETURN
      END IF
      '
      LOCATE 10, 20: INPUT "DUMP TO SCREEN (1) OR PRINTER (2) "; IANSW
      IF IANSW = 1 THEN 28610
      IF IANSW > 2 OR IANSW < 1 THEN 29110
      LPRINT CHR$(18)
      LPRINT STRING$(60, "-"): LPRINT DATE$; "    "; TIME$
      LPRINT STRING$(60, "-"): LPRINT
      PRINT : PRINT "NUMBER OF MISSIONS : "; NUMMSNS: PRINT
29180 INPUT "ENTER SEQUENCE NUMBER OF FIRST MISSION TO PRINT ";JFIRST
      IF JFIRST < 1 OR JFIRST > NUMMSNS THEN BEEP: GOTO 29180
29200 INPUT "ENTER SEQUENCE NUMBER OF LAST MISSION TO PRINT "; JLAST
      IF JLAST < JFIRST OR JLAST > NUMMSNS THEN BEEP: GOTO 29200
```

```
      FOR I = JFIRST TO JLAST
         LPRINT
          IZTME = IMST(I, ICTM): GOSUB 29600
          IXTME = IMST(I, ICTM) + LEGS(IMST(I, IFSTLG), 2)
          IXTME = INT(IXTME / 1440)
          MID$(SMST(I), 10, 3) = RIGHT$(STR$(JDLA + IXTME + 1000), 3)
          TEMP = VAL(RIGHT$(SMST(I), 3))
          IF TEMP <= 365 AND TEMP > 0 THEN 29340
          IF TEMP = 0 THEN TEMP = 365 + LSTLEAP: GOTO 29320
          TEMP = TEMP - 365 + LSTLEAP
29320     MID$(SMST(I), 10, 3) = RIGHT$(STR$(TEMP + 1000), 3)
          '
29340     LPRINT "FLOW SEQUENCE #  "; I; "     "; SAFLD(JCB, 2);
          LPRINT "  DEPARTURE TIME "; STME
          LPRINT "ACFT "; SJETS(IMST(I, IJET)); "     ";
          LPRINT ISQUAD(IMST(I, ISQD), 1); " TAS ";
          LPRINT "      MISSION #  "; SMST(I): LPRINT
          LPRINT "ARRIVE      GNDTM          DEPART        FLTTM   LEG"
          '
          FOR J = IFSTLG TO ILSTLG
             IF J <> IFSTLG THEN ZFRM = "": GOTO 29420
             IF IMST(I, ILKFRM) <> 0 THEN
               ZFRM = " CYCLES FROM " + SMST(IMST(I, ILKFRM))
             ELSE
               ZFRM = ""
             END IF
             IF IMST(I, ILKTO) <> 0 THEN
               ZTO = " CYCLES TO " + SMST(IMST(I, ILKTO))
             ELSE
               ZTO = ""
             END IF
29420        IF IMST(I, J) = 0 THEN J = ILSTLG: GOTO 29520
             ITMP2 = LEGS(IMST(I, J), 4): GOSUB 29770
             '
            IZTME = IMST(I,ICTM) + LEGS(IMST(I,J),2): GOSUB 29600
             IF J = IFSTLG THEN LPRINT SPC(23);
             LPRINT SAFLD(LEGS(IMST(I, J), 3), 2); "     "; STME; "  ";
             LPRINT STME2; "     "; IMST(I, J)
             ITMP2 = LEGS(IMST(I, J), 5): GOSUB 29770
             IZTME = IZTME + LEGS(IMST(I, J), 4): GOSUB 29600
             LPRINT STME + "  " + STME2 + "     ";
             IF IMST(I, J + 1) = 0 OR J = ILSTLG THEN
               LPRINT " "; SAFLD(LEGS(IMST(I, J), 6), 2)
               LPRINT ZTO
             END IF
29520     NEXT J
          LPRINT : LPRINT ZTO
      NEXT I
      RETURN     ' to 650
```

```
'
'    time conversion
'    called from 28610,28740,28950,28990,29240,29450, 29490
'
29600 LSTFLG = 0
      LTME = IZTME
      KDY = INT(LTME / 1440)
      JULIAN = KDY + JDLA
      IF JULIAN <= 0 THEN
        LSTFLG = 1
         JULIAN = JULIAN + 365 + LSTLEAP
      ELSEIF JULIAN > 365 + LEAP THEN
         JULIAN = JULIAN - (365 + LEAP)
      END IF
      '
      GOSUB 29980       'IDT,ZMO,ZYR
      LTME = LTME - KDY * 1440
      IHOUR = INT(LTME / 60) + 100
      IMIN = LTME - (IHOUR - 100) * 60 + 100
      T$ = RIGHT$(STR$(IHOUR), 2) + RIGHT$(STR$(IMIN), 2)
      IF T$ = "0000" THEN T$ = "0001"
      STME = ZDATE + " " + T$
      RETURN
      '
      '  minutes to hrs+min
      '  called from 28840, 28980, 29430, 29480
      '
29770 IHR = INT(ITMP2 / 60): IMN = ITMP2 MOD 60
      STME2 = RIGHT$(STR$(IHR), 2) + "+" + RIGHT$(STR$(IMN + 100), 2)
      IF STME2 = " 0+00" THEN STME2 = "    "
      RETURN
      '
      '  calendar to julian
      '  called from 23390
      '
29850 BADFLG = 0
      IDY = VAL(LEFT$(SDATE, 2))
      IMO = VAL(MID$(SDATE, 3, 2)): IYR = VAL(RIGHT$(SDTE, 2))
      NYR = VAL(RIGHT$(SDATE, 2))
      IF IDY > 31 OR IMO > 12 OR IDY < 1 OR IMO < 1 THEN
        BADFLG = 1
         RETURN
      END IF
      IF IYR MOD 4 = 0 THEN LEAP = 1 ELSE LEAP = 0
      IF IYR MOD 4 = 1 THEN LSTLEAP = 1 ELSE LSTLEAP = 0
      JULIAN = NDATES(IMO) + IDY
      IF NYR < IYR THEN IOFFSET = LSTLEAP ELSE IOFFSET = LEAP
      IF JULIAN > 59 THEN JULIAN = JULIAN + IOFFSET
      RETURN
      '
```

```
'    julian to calendar
'    called from 29650
'
29980 JFLG = 0: DTFLG = 0
      IF JULIAN = 60 AND LEAP = 1 THEN '  feb 29
         ZDATE = "29 FEB"
         RETURN
      END IF
      IF JULIAN = 366 THEN JFLG = 1
      IF JULIAN > 59 THEN JULIAN = JULIAN - LEAP
      FOR L = 2 TO 12
          IF NDATES(L) = JULIAN THEN
            LL = L
            EXIT FOR
          END IF
          IF NDATES(L) > JULIAN THEN LL = L: EXIT FOR
      NEXT L
      IF JULIAN > 334 THEN LL = 13    ' leap december
30040 IF LSTFLG = 1 THEN LSTFLG = LSTLEAP
      JDY = JULIAN - NDATES(LL - 1) - LSTFLG
      SMO = MID$(ZMONTH, ((3 * (LL - 1)) - 2), 3)
      STEMP2 = STR$(JDY + 100)
      ZDATE = RIGHT$(STEMP2, 2) + " " + SMO
      IF JULIAN >= 59 THEN JULIAN = JULIAN + LEAP
      RETURN     ' to 29650
      '
      '  calculate great circle distance
      '
30920 RAD = 57.2958
      RPI = 3.14159
      RLAT1 = RLAT1 / RAD: RLONG1 = RLONG1 / RAD
      RLAT2 = RLAT2 / RAD: RLONG2 = RLONG2 / RAD
      '
      ACSN1 = SIN(RLAT1) * SIN(RLAT2)
      ACSN2 = COS(RLAT1) * COS(RLAT2) * COS(ABS(RLONG2 - RLONG1))
      ACSN = ACSN1 + ACSN2
      ACSD = 1 - (ACSN ^ 2)
      IF ACSD = 0 THEN DIST = 0: GOTO 30950
      ACOS = (RPI / 2) - (ATN(ACSN / SQR(ACSD)))
      DIST = INT(3439.77 * ACOS)
30950 RETURN
      '
```

```
      ' prioritize missions
      '
40000 CLS
      LOCATE 10, 20
      PRINT "THIS WILL ERASE THE EXISTING FLOWPLAN."
      LOCATE 12, 20: INPUT "'C' TO CONTINUE, 'ENTER' TO EXIT  "; SANSW
      IF SANSW <> "C" AND SANSW <> "c" THEN RETURN
      '
      NSQDEP = 0
      FOR I = 1 TO NSQUAD
          FLOWTEMP(I) = ISQUAD(I, 4)        ' paa
      NEXT I
      '
      FOR I = 1 TO MAXPLANS
          NSQDPLY(I, 1) = 0: NSQDPLY(I, 2) = 0
      NEXT I
      '  print templates
      FOR II = 1 TO MAXPLANS
          CLS
          FOR I = (MAXMSNS + 1) TO (MAXMSNS + MAXPLANS)
             Q = I - MAXMSNS: Y = Q
             LOCATE Y, 5
             IF IMST(I, IJET) = 0 THEN 40010
             PRINT Q; "  "; : IF Q < 10 THEN PRINT " ";
             PRINT SJETS(IMST(I, IJET)); "  ";
             SQUADRON=RIGHT$("    "+STR$(ISQUAD(IMST(I,ISQD), 1)), 4)
             PRINT SQUADRON; " TAS  ";
             FOR J = IFSTBSE TO ILSTBSE
                IF IMST(I, J) = 0 OR J = ILSTBSE THEN
                   LOCATE Y, 65
                   PRINT SNOTE(Q)
                   EXIT FOR
                END IF
                PRINT SAFLD(INT(IMST(I, J) / 100), 2); " ";
             NEXT J
40010     NEXT I
          '
          FOR I = 1 TO MAXPLANS
           IF NSQDPLY(I,1) <> 0 THEN LOCATE NSQDPLY(I,1),1 :PRINT"->"
          NEXT I
          '
```

```
          LOCATE Y + 2, 15
40020     INPUT"TEMPLATE YOU WANT TO USE 'RETURN' to continue "; IANSW
          IF IANSW <= 0 THEN EXIT FOR
           IF IANSW > NUMTEMPS THEN BEEP: GOTO 40020
           '
           WFLAG = 0
           JSQUAD = IMST(MAXMSNS + IANSW, ISQD)
           IF FLOWTEMP(JSQUAD) = 0 THEN     ' sq has no more acft
             LOCATE Y + 1, 20
             PRINT SPC(59); ""
             LOCATE Y + 2, 15
             PRINT "SQUADRON CHOSEN HAS NO MORE ACFT TO DEPLOY"
             LOCATE Y + 3, 20: INPUT "      'ENTER TO CONTINUE "; SS
             LOCATE Y + 2, 15: PRINT SPC(50); ""
             LOCATE Y + 3, 20: PRINT SPC(50); ""
             LOCATE Y + 2, 15: GOTO 40020
           END IF
           '
          NSQDPLY(NSQDEP + 1,1) = IANSW    'which template is chosen
           NSQDEP = NSQDEP + 1              'how many templates chosen
           '
           TMAX = FLOWTEMP(JSQUAD)
           PRINT "           NUMBER OF ACFT FROM THIS SQUADRON";
           PRINT " ( 1 - "; TMAX; ")"
40030     LOCATE Y + 3, 56: INPUT IANSW2
           IF IANSW2 < 0 OR IANSW2 > TMAX THEN
             LOCATE Y + 3, 55: PRINT "          "
             GOTO 40030
           END IF
           IF IANSW2 = 0 THEN
             NSQDPLY(II, 1) = 0
             NSQDEP = NSQDEP - 1
             LOCATE Y + 2, 15: PRINT SPC(60); ""
             LOCATE Y + 3, 10: PRINT SPC(65); ""
             LOCATE Y + 2, 15: GOTO 40020
           ELSE
             NSQDPLY(II, 2) = IANSW2          ' how many acft deployed
             FLOWTEMP(JSQUAD) = FLOWTEMP(JSQUAD) - IANSW2
          END IF
       NEXT II
       CLS : LOCATE 10, 34: PRINT "STANDBY . . ."
       IF NSQDEP = 0 THEN RETURN
```

```
'
'  sort by lad
'
FOR I = 1 TO NSQDEP - 1
   FOR J = 1 TO NSQDEP - 1
       ITEMP1 = ISQUAD(IMST(MAXMSNS + NSQDPLY(J, 1), ISQD), 6)
       ITEMP2 = ISQUAD(IMST(MAXMSNS + NSQDPLY(J + 1,1) ,ISQD),6)
      IF ITEMP1 > ITEMP2 THEN
         SWAP NSQDPLY(J, 1), NSQDPLY(J + 1, 1)
       END IF
    NEXT J
NEXT I
'
'     load missions into IMST array
'
ALPHA = 0: OMEGA = 0: NUMMSNS = 0
FOR I1 = 1 TO NSQDEP
    ALPHA = OMEGA + 1
    OMEGA = OMEGA + NSQDPLY(I1, 2)
    FOR J1 = ALPHA TO OMEGA
       FOR K1 = 1 TO IMSTTOP
          IMST(J1, K1) = IMST(MAXMSNS + NSQDPLY(I1, 1), K1)
       NEXT K1
       SMST(J1) = SMST(MAXMSNS + NSQDPLY(I1, 1))
    NEXT J1
NEXT I1
NUMMSNS = OMEGA : ALPHA = 1
GOSUB 50000    ' clear LEGS array
GOSUB 21010    ' flow & mog
RETURN
'
50000 FOR IJ = 1 TO MAXLEGS
       FOR II = 1 TO 7
          LEGS(IJ, II) = 0
       NEXT II
NEXT IJ
RETURN
```

```
60000 '
      ' change squadron data
      '
      CLS : INUSQ = 0
      LOCATE 8, 20
      PRINT "1       <-  TO CHANGE DATA FOR AN EXISTING SQUADRON"
      LOCATE 10, 20: IF MAXADDSQ = 0 THEN 60001
      PRINT "2       <-  TO ADD A NEW SQUADRON ("; MAXADDSQ; "MAX )"
      LOCATE 12, 17
60001 PRINT "'ENTER'    <-  TO RETURN TO MAIN MENU"
      LOCATE 15, 20: INPUT IANSW
      IF IANSW = 2 AND MAXADDSQ = 0 THEN 60000
      IF IANSW < 0 OR IANSW > 2 THEN 60000
      IF IANSW = 0 THEN RETURN
      IF IANSW = 2 THEN
          INUSQ = 1: GOSUB 60010: GOSUB 60020: GOSUB 60030
          GOSUB 60040: GOSUB 60050: GOTO 60005
      END IF
      '
60003 CLS : LOCATE 10, 20: INPUT "ENTER SQUADRON . . ."; IANSW0
      IOK = 0
      FOR I = 1 TO NSQUAD
          IF ISQUAD(I, 1) = IANSW0 THEN IOK = 1
      NEXT I
      IF IOK = 0 THEN 60003
      '
60005 CLS : GOSUB 60100
      FOR I = 1 TO NSQUAD
          IF ISQUAD(I, 1) = IANSW0 THEN
            ISQPTR = I
            LOCATE 6, 13: PRINT ISQUAD(I, 1)
            LOCATE 6, 27: PRINT SAFLD(ISQUAD(I, 5), 2)
            LOCATE 6, 39: PRINT SERVICE(ISQUAD(I, 2))
            LOCATE 6, 53: PRINT SJETS(ISQUAD(I, 3))
            LOCATE 6, 66: PRINT ISQUAD(I, 4)
            LOCATE 13, 29: PRINT ISQUAD(I, 6)
            LOCATE 13, 41: PRINT ISQUAD(I, 7)
          END IF
      NEXT I
      IF INUSQ = 1 THEN 60008
      LOCATE 18, 20: PRINT "ENTER NUMBER OF FIELD TO CHANGE, OR"
60008 LOCATE 20, 20: INPUT "'ENTER' TO CONTINUE"; IANSW1
      IF INUSQ = 1 OR IANSW1 = 0 THEN RETURN
      IF IANSW1 = 1 OR IANSW1 > 7 THEN BEEP: GOTO 60005
      ICHGSQ = 1   ' triggers save option when ending
      ON IANSW1 GOSUB 60010,60020,60030,60040,60050,60060, 60070
      GOTO 60005
      '
```

```
60010 CLS : IOK = 0
      LOCATE 10, 20: INPUT "ENTER NEW SQUADRON ID"; IANSWO
      IF IANSWO = 0 THEN 60010
      FOR I = 1 TO NSQUAD
          IF ISQUAD(I, 1) = IANSWO THEN IOK = 1
      NEXT I
      IF IOK = 1 THEN
         LOCATE 12, 20
         INPUT "SQUADRON ALREADY EXISTS.  'ENTER' TO CONTINUE "; SS
        GOTO 60010' to avoid repeating sq
      END IF
      NSQUAD = NSQUAD + 1
      ISQUAD(NSQUAD, 1) = IANSWO
      ISQPTR = NSQUAD
      ICHGSQ = 1     ' triggers save option when ending
      MAXADDSQ = MAXADDSQ - 1
      RETURN
      '
60020 CLS : LOCATE 12, 20
      INPUT "ENTER ICAO IDENTIFIER OF HOME BASE"; SANSW
      IOK = 0: SS = ""
      FOR I = 1 TO AFLDS
          IF SAFLD(I, 2) = SANSW THEN ISQUAD(ISQPTR, 5) = I: IOK = 1
      NEXT I
      IF IOK = 0 THEN
         LOCATE 12, 20: PRINT SPC(60); SS
         LOCATE 12, 20
         INPUT "ICAO NOT IN DATA BASE. 'ENTER' TO CONTINUE"; SS
         LOCATE 12, 20: PRINT SPC(60); SS
         GOTO 60020
      END IF
      RETURN
      '
60030 CLS : Y = 8
      FOR I = 1 TO 3
         LOCATE Y, 20: PRINT I, SERVICE(I)
          Y = Y + 2
      NEXT I
      LOCATE Y + 2, 20
      INPUT"ENTER NUMBER FOR TYPE SQUADRON"; ISQUAD(ISQPTR, 2)
      RETURN
      '
60040 CLS : Y = 1
      FOR I = 1 TO NUMACFT
          LOCATE Y, 20: PRINT I, SJETS(I)
          Y = Y + 2
      NEXT I
      PRINT : PRINT : PRINT SPC(20);
      INPUT"ENTER NUMBER FOR ASSIGNED ACFT";ISQUAD( ISQPTR,3)
      RETURN
      '
```

```
60050 PRINT : PRINT SPC(20);
      INPUT "ENTER NUMBER OF AIRCRAFT AT BASE"; IANSW2
      ISQUAD(ISQPTR, 4) = IANSW2
      RETURN
      '
60060 INPUT "ENTER DAY SQUADRON IS AVAILABLE (1, 2, 3, etc)";IANSW3
      ISQUAD(ISQPTR, 6) = IANSW3
      RETURN
      '
60070 INPUT "ENTER SQUADRON'S DEPLOYMENT DEADLINE (1, 2, 3, etc)";IANSW4
      ISQUAD(ISQPTR, 7) = IANSW4
      RETURN
      '
60100 CLS : SS = "    ": PRINT
      PRINT "FIELD  -->";
      PRINT "   1              2             3             4             5"
      IF IANSW = 1 THEN LOCATE 2, 12: PRINT SS
      PRINT : PRINT SPC(10);
      PRINT "SQUADRON        BASE          TYPE          ACFT          PAA"
      PRINT SPC(10);
      PRINT "--------        ----          ----          ----          ---"
      PRINT
      LOCATE 9, 1: PRINT "FIELD  -->"
      LOCATE 9, 30: PRINT "6            7"
      LOCATE 11, 26: PRINT "AVAILABLE    IN PLACE"
      LOCATE 12, 26: PRINT "---------    --------"
      RETURN
      '
61000 IF ICHGSQ = 1 THEN
         LOCATE 10, 20
         INPUT "SAVE CHANGES MADE TO SQUADRON DATABASE (Y/N)  "; SANSW
         IF SANSW = "Y" OR SANSW = "y" THEN
          OPEN "O", 2, "SQUAD#.DAT"
           PRINT #2, NSQUAD
           FOR I = 1 TO NSQUAD
              FOR J = 1 TO 7
                  IF J = 5 THEN
                    PRINT #2, SAFLD(ISQUAD(I, J), 2)
                  ELSE
                    PRINT #2, STR$(ISQUAD(I, J))
                  END IF
              NEXT J
           NEXT I
         END IF
      END IF
      '
```

```
IF ICHBSE = 1 THEN
   CLS : LOCATE 10, 20
   INPUT 'SAVE CHANGES MADE TO AIRFIELD DATABASE (Y/N)  '; SANSW
   IF SANSW = 'Y' OR SANSW = 'y' THEN
     OPEN 'O', 2, 'BASE#.DAT'
     PRINT #2, AFLDS
     FOR I = 1 TO AFLDS
        PRINT #2, SAFLD(I, 1)
        PRINT #2, SAFLD(I, 2)
        FOR J = 1 TO 4
            PRINT #2, STR#(AFLD(I, J))
        NEXT J
      NEXT I
   END IF
END IF
'
END     'program
```

```
      '
62000 '     add airfield to data base
      '
      CLS
      IF AFLDS = MXAFLD THEN
         LOCATE 10, 10
         PRINT "YOU MUST RESTART THE PROGRAM TO ADD MORE AIRFIELDS"
         LOCATE 12, 25: INPUT "'ENTER' TO CONTINUE"; SS
         GOTO 62999
      END IF
      LOCATE 4, 1
      PRINT "'ENTER' FOR ICAO TO ABORT.": PRINT
      PRINT "ENTER LAT & LONG IN DEGREES AND TENTHS.": PRINT
      PRINT "ENTER SOUTH LATITUDE AND EAST LONG AS NEGATIVE"
      LOCATE 9, 3
      PRINT "_____."
      LOCATE 11, 3
      PRINT "ICAO   NAME          LAT     LONG     NARROW MOG   WIDE MOG"
      LOCATE 13, 3
      PRINT "____   ______________  ____   _____     __  __."
      I = AFLDS + 1
      LOCATE 12, 1: INPUT STEST
      '    abort test
      IF STEST = "" THEN 62999
      '    in database test
      FOR J = 1 TO AFLDS
          IF SAFLD(J, 2) = STEST THEN
            CLS : LOCATE 10, 20: PRINT "AIRFIELD ALREADY IN DATA BASE."
            PRINT SPC(19); : INPUT "'ENTER' TO CONTINUE"; SS
            GOTO 62999
          ELSE
           SAFLD(I, 2) = STEST
          END IF
      NEXT J
      LOCATE 12, 8: INPUT SAFLD(I, 1)
      LOCATE 12, 24: INPUT AFLD(I, 1)
      LOCATE 12, 32: INPUT AFLD(I, 2)
      LOCATE 12, 45: INPUT AFLD(I, 3)
      LOCATE 12, 59: INPUT AFLD(I, 4)
      GOSUB 4270: IF SANSW = "N" OR SANSW = "n" THEN 62000
      ICHBSE = 1
      AFLDS = I
62999 RETURN
```

BIBLIOGRAPHY

1. Abreu, Maj. Michael H. and Pritchard, Russell A. report number CAA-D-87-11, U.S. Army Concepts Analysis Agency, Bethesda, MD, September 1987.

2. Bowditch, Nathaniel. American Practical Navigator (1977 Edition), Volume I. Defense Mapping Agency Hydrographic Center, DMA stock number NVPUB9V1

3. Crossman, Maj. Pete. 436 Military Airlift Wing, Dover AFB, DE, 1984.

4. Cuda, Capt Daniel L. Dynamics of Airfield Parking Congestion During APOD Operations. MS Thesis, GST/OS/85M-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1985 (AD A155 778).

5. Foster, Capt. Mike; Farwell, Capt. Dean; Barber, 1Lt Doug. 60 Military Airlift Wing, Travis AFB, CA, August 1984.

6. Hill, Maj. James E. HQ MAC/XP-CAAG, Scott AFB, Il, telephone interview, 6 November 1987.

7. Hill, Maj. James E. and Donnelly, Capt Mark S. Limitations To European C-130 Deployment. MS Thesis, GST/OS/86M-5. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH March 1986 (AD A172 527).

8. Jones, Maj. Brian. Hq MAC/XOSP, Scott AFB, IL, Telephone and personal conversations, October-November 1987.

9. Perry, Maj. Greg. Airlift Operations School, HQ MAC/AOS Scott AFB, IL, personal and telephone interviews, October-November 1987.

10. Prosser, Maj. Terry. Hq MAC/XOSP, Scott AFB, Il, telephone interviews 7 and 9 October 1987.

11. SAS Institute, Inc. SAS/GRAPH USER'S GUIDE, (Version 5 Edition). Cary, NC: SAS Institute, Inc., 1985

12. Wilson, Lt Col Warren. Hq MAC/XOSP, Scott AFB, Il, telephone interview, 13 November 1987 and correspondence, December 1987.

VITA

Major Mike Foster was born on 2 June 1950 in Bristow, Oklahoma. He graduated from high school in Midwest City, Oklahoma, in 1968 and attended the University of Oklahoma, from which he received the degree of Bachelor of Arts in Anthropology in May 1974. After receiving his commission from the Officer Training School in January, 1975, he attended Undergraduate Navigator Training and F-4 training. He served as a Weapon System Officer in the 512th Tactical Fighter Squadron, Ramstein AB, Federal Republic of Germany. He completed pilot training and received his wings in April, 1979, and was an instructor pilot and flight evaluator in the 323rd Flight Training Wing, Mather AFB, CA from April 1979 until January 1983. In August, 1982, he received the degree of Bachelor of Science in Computer Science from Chapman College. He served as a C-5 aircraft commander in the 60th Military Airlift Wing, Travis AFB, CA, until entering the School of Engineering, Air Force Institute of Technology, in September, 1986.

Permanant address: 4120 Quail Bush Drive

Dayton, Ohio 45424

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: School of Engineering / 6b OFFICE SYMBOL (If applicable): AFIT/ENS	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code): Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio 45433-6583	7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING SPONSORING ORGANIZATION / 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
DEPLOY: An Airlift Scheduler For Real-Time Crisis Action Planning

12 PERSONAL AUTHOR(S)
R. Mike Foster, B.A., B.S., Maj, USAF

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
MS Thesis	FROM _____ TO _____	1988 March	123

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Airlift Operations, Deployment, Scheduling
15	05		
15	06		

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advisors: Thomas F. Schuppe, Lt Col, USAF
Instructor of Operations Research

John R. Valusek, Lt Col, USAF
Assistant Professor of Operations Research

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Thomas F. Schuppe, Lt Col, USAF	(513)-255-3362	AFIT/ENS

DD Form 1473, JUN 86 — Previous editions are obsolete — SECURITY CLASSIFICATION OF THIS PAGE

Preparation for a war in Europe would include the deployment of C-130 tactical airlift aircraft from their home bases in the United States to beddown locations in Europe. Methods currently used to plan such deployments would require as long as two days to provide a complete deconflicted deployment schedule.

The purpose of this study was to automate the deployment planning process. A review of literature concentrated on recent theses which studied airlift problems and existing deployment planning software.

An interactive program was written, based on the concept of using departure time from refueling choke points as the mechanism for regulating aircraft flow. A deployment flow plan for all active duty C-130s stationed in the U.S can be completed, using this program, in approximately one hour.

END
DATE
FILMED
5-88
DTIC